



HEINZ NIXDORF INSTITUT
Universität Paderborn
Wirtschaftsinformatik, insbesondere CIM

Geregelte Vereinfachung hierarchischer Partitionen von Modellen in der Materialflusssimulation

Dissertation
zur Erlangung der Würde eines Doktors
der Wirtschaftswissenschaften (Dr. rer. pol.)
der Universität Paderborn

vorgelegt von
Dipl.-Wirt.-Ing. Daniel Huber
33098 Paderborn

Paderborn, Oktober 2009

Dekan: Prof. Dr. Peter F. E. Sloane
Referent: Prof. Dr.-Ing. habil. Wilhelm Dangelmaier
Korreferent: Prof. Dr. Leena Suhl

Tapferkeit
Aufopferung
Mitleid
Gerechtigkeit
Ehrlichkeit
Ehre
Demut

Mein besonderer Dank gilt:

Philip Hartmann

Tobias Horstmann

David Janßen

Oskar Kunst

Daniel Müller

Allesamt Studenten,
die mich mit Rat und Tat
bei der Erstellung dieser Arbeit unterstützt haben.

Inhaltsverzeichnis

Abbildungsverzeichnis	v
Tabellenverzeichnis	vii
Abkürzungsverzeichnis	ix
1 Einleitung	1
1.1 Motivation zum Einsatz vereinfachter Modelle in der Materialflusssimulation	1
1.2 Ziele und Vorgehensweise	2
2 Problemstellung	5
2.1 Problemdefinition	5
2.1.1 Modelle in der Materialflusssimulation	6
2.1.2 Vereinfachung von Materialflussmodellen	7
2.1.3 Die detaillierungsvariante diskrete Simulation	9
2.1.4 Partitionierung und Hierarchisierung	13
2.1.5 Regelung der Vereinfachung	14
2.1.6 Konsistenz und Qualität des Simulationszustands nach Rekombination	15
2.1.7 Integration des Verfahrens in das Simulationswerkzeug d ³ FACT . .	17
2.2 Anforderungen an die Problemlösung	17
2.2.1 Anforderungen an die Modellbeschreibung	18
2.2.2 Anforderungen an die Partitionierung und Hierarchisierung	19
2.2.3 Anforderungen an die automatisierte Modellvereinfachung	21
2.2.4 Anforderung an die Regelung des Vereinfachungsalgorithmus	21
2.2.5 Anforderungen an die Generierung von Zustandsabbildungen	22
2.2.6 Anforderungen aus der Integration des Verfahrens in d ³ FACT . . .	23
3 Stand der Technik	25
3.1 Modellbeschreibungsmethoden in der Materialflusssimulation	25
3.1.1 Modellbeschreibungsmethoden im Allgemeinen	25
3.1.2 Die Modellbeschreibungsmethode der dvdS	26
3.1.3 Die Modellbeschreibungsmethode von d ³ FACT	28
3.1.4 Die Modellbeschreibungsmethode der Discrete Event System Spe-	
cification	29
3.2 Methoden der Partitionierung und Hierarchisierung	30
3.2.1 Partitionierungsalgorithmen	30
3.2.2 Methoden der Hierarchisierung	35

3.3	Methoden der Vereinfachung von ereignisdiskreten Simulationsmodellen . .	36
3.3.1	Grundlegende Ziele und Methoden	37
3.3.2	Benutzung von Homomorphismen zur Vereinfachung	40
3.3.3	Gezielte Reduzierung der Zahl ausmodellierter Anlagen oder Arbeitsgänge	41
3.3.4	Einsatz von Simulationsmetamodellen	47
3.4	Methoden zum Messen der Komplexität und Verhaltensabweichung	48
3.4.1	Messen der Komplexität eines Modells	49
3.4.2	Messen des Verhaltens und Validieren eines Modells	51
3.5	Methoden zum Erzeugen und Verändern von Zuständen	52
3.5.1	Zustände im Bereich Parallele und Verteilte Simulation	52
3.5.2	Anforderungen an die Konsistenz beim Austausch von Teilmodellen	54
3.5.3	Erzeugung von konsistenten Zuständen in der dvdS	55
3.6	Die Simulationsplattform	56
3.6.1	Benutzerstimulierte detaillierungsvariante Simulation	57
3.6.2	Rückwärtssimulation	58
3.6.3	Sonstige Module von d ³ FACT	60
4	Zu leistende Arbeit	69
5	Konzeption	73
5.1	Übersicht	73
5.2	Konzept einer Modellbeschreibungsmethode	76
5.2.1	Festlegung der Menge von Modellkomponentenklassen	76
5.2.2	Modellbeschreibung	84
5.3	Konzept einer Partitionierungsmethode	84
5.4	Konzeption von Methoden zur automatisierten Modellvereinfachung	86
5.4.1	Zusammenfassung von Modellkomponenten	87
5.4.2	Transformation von Variablen	102
5.4.3	Weglassung von Komponenten	103
5.5	Konzeption von Regelungstechniken der Vereinfachung	108
5.5.1	Erstellung der Zielvorgaben	109
5.5.2	Komplexitätsmaß	110
5.5.3	Messung Verhaltensabweichung	113
5.5.4	Algorithmus der Regelung	116
5.6	Konzepte zur Generierung von Zustandsabbildungsfunktionen	119
5.6.1	Grundlagen der Zustandsgenerierung	119
5.6.2	Bestimmung der Belegung der zu aktivierenden Teilmodelle	121
5.6.3	Bestimmung der Störstatus der zu aktivierenden Teilmodelle	123
5.6.4	Bestimmung der Statistikvariablen der zu aktivierenden Teilmodelle	124
6	Validierung	127
6.1	Übersicht	127
6.2	Testmodelle	129

6.3	Validierung der Partitionierungsmethode	132
6.3.1	Ergebnisse	132
6.3.2	Bewertung der Ergebnisse	133
6.4	Validierung der Vereinfachungsmethode	137
6.4.1	Ergebnisse	137
6.4.2	Bewertung der Ergebnisse	139
6.5	Validierung der Regelung	140
6.5.1	Ergebnisse	141
6.5.2	Bewertung der Ergebnisse	144
6.6	Validierung der Zustandsgenerierungsmethode	145
6.6.1	Ergebnisse	146
6.6.2	Bewertung der Ergebnisse	147
7	Zusammenfassung und Ausblick	149
7.1	Ergebnis	150
7.2	Weiterer Forschungsbedarf	152
	Literaturverzeichnis	155
	Stichwortverzeichnis	163

Abbildungsverzeichnis

2.1	Aufbau eines dvdS-Modells	11
2.2	Rekomposition eines dvdS-Modells	12
3.1	Ein Beispiel für das System Entity Structure/Model Base Framework	36
3.2	Taxonomie von Vereinfachungsmethoden [Fra95]	38
3.3	Konsistenzanforderungen an die Umschaltvorgänge	54
3.4	Strukturerkennung zur iterativen Zusammenfassung.	60
3.5	d ³ FACT Modellierungsskizze	61
3.6	Modellstruktur für die detaillierungsvariante Simulation	64
3.7	d ³ FACT Bildschirmfotos	66
5.1	Verfahrensübersicht des Gesamtsystems	74
5.2	Regelung der Vereinfachung anhand Komplexität und Validität	75
5.3	Mehrere Sequenzer in zusammenzufassender Linie	94
5.4	Mehrere Sequenzer in zusammenzufassender Linie	100
5.5	Zeitmessung für Bestimmung von Zufallsvariablen	103
5.6	Zielvorgaben der Vereinfachung (Beispiel)	109
5.7	Komplexitäts- Verhaltensabweichungsverlauf für eine Vereinfachung (Skizze)	117
6.1	Struktur und Materialflussrichtung Modell A	130
6.2	Struktur und Materialflussrichtung Modell B	131
6.3	Struktur und Materialflussrichtung Modell C	131
6.4	Partitionierungen Modell A	134
6.5	Partitionierungen Modell B	135
6.6	Partitionierungen Modell C	135
6.7	Partitionierungen mit schlechtem Ergebnis	136
6.8	Komplexität im Verlauf der Vereinfachung	138
6.9	Komplexität über Verhaltensabweichung von vereinfachten Modellen	141
6.10	Komplexitätsmaß und Laufzeit von Modell A	143
6.11	Maximale Vereinfachung Modell C	145

Tabellenverzeichnis

3.1	Fehleranalyse zwischen Ausgangsmodell und Vereinfachungen [HL99] . . .	44
3.2	Reduzierung der Laufzeit durch Vereinfachung [HL99]	45
3.3	Übersicht Übergangsfunktionen [Mue05]	56
3.4	Übersicht Grundstrukturen	58
5.1	Zusammenfassung von zwei oder mehr Maschinen Normal in einer Linie . .	92
5.2	Zusammenfassung von zwei oder mehr Förderbändern in einer Linie	92
5.3	Zusammenfassung von Puffern in einer Linie	93
5.4	Zusammenfassung von gemischten Linien	95
5.5	Zusammenfassung einer Montagemaschine mit einer gemischten Linie . . .	96
5.6	Zusammenfassung einer Demontagemaschine mit einer gemischten Linie . .	97
5.7	Zusammenfassung homogener Linien	99
5.8	Verwendete \mathcal{C}_v	111
6.1	Versuchsplan	128
6.2	Bestimmung des maximalen Ratio-Cuts	133
6.3	Variationskoeffizienten des Ratio-Cuts	133
6.4	Erhalt der Stochastik	139
6.5	Güte der Engstellenberechnungsfunktion	139
6.6	Korrelationsmessung zwischen Maß und Laufzeit Modell A	142
6.7	Korrelationsmessung zwischen Maß und Laufzeit Modell B	142
6.8	Korrelationsmessung zwischen Maß und Laufzeit Modell C	142
6.9	Abhängigkeit der Verhaltensabweichung von der Auslastung	144
6.10	Abweichung des Durchsatzes nach Umschalten von Teilmodellen	146
6.11	Fehler beim Umschalten von Teilmodellen ohne zwischenzeitliche Simulation	147

Abkürzungsverzeichnis

dvdS	Detaillierungsvariante diskrete Simulation
dvdS-Modell	Ein aus Teilmodellen unterschiedlicher Komplexität zusammengesetztes Modell, welches das gesamte System abbildet
F	Förderband
M	Normale Maschine
MD	Demontagemaschine
MM	Montagemaschine
PF	Fifo-Puffer
PL	Lifo-Puffer
PP	Prioritätspuffer
Q	Quelle
QI	Quelle Intervall
QL	Quelle Liste
QV	Quelle Verteilung
S	Senke
SQ	Sequenz
SS	Senke mit Statistikauswertung
Ω	Das Ausgangsmodell
VB	Verteiler mit Beschreibungs-Methode
VR	Verteiler mit Reihum-Methode
VV	Verteiler mit Verteilungs-Methode
Z	Zusammenführung
t_U	Bearbeitungszeit
LT	Durchlaufzeit
TP	Durchsatz
BN	Menge von Engpässen
bn	Engpass
E	Menge der Ereignisse
e	Ereignis
\mathcal{H}	Anzahl der zu erzeugenden Hierarchieebenen
H	Menge der Hierarchieebenen
h	Eine Hierarchieebene

C	Kapazität
\mathcal{C}	Komplexität
$\mathcal{C}_Z(h)$	Zielvorgabe über die zu erreichende Komplexität auf Hierarchieebene h
D	Menge der Komponentenklassen
d	Eine Komponentenklasse
V	Menge der Komponenten
v	Eine Komponente
\mathcal{L}	Laufzeit
LS	Losgröße
M	Modell
$MTTF$	Mean Time to Failure
$MTTR$	Mean Time To Repair
NN	Neuronales Netz
K_P	Menge der Kanten in der Hierarchie der Partitionen
k_p	Eine Kante zwischen zwei Partitionen in der Hierarchie der Partitionen
p	Partition
$p_{i,h}$	Partition i auf Hierarchieebene h
P	Menge der Partitionen
O	Menge der Komponentenpositionen
o	Eine Komponentenposition
BS	Puffergröße
t_R	Rüstzeit
\mathcal{S}	Geschnittene Kantenmenge
ϕ	Struktur der Vereinfachung
$\Phi(b)$	Menge von Strukturen in Teilmodell b
\check{B}	Menge der inaktiven Teilmodelle
\hat{B}	Menge der aktiven Teilmodelle
$b_{i,h}$	Teilmodell i auf Hierarchieebene h
$B_j(b_{i,h})$	Menge der Teilmodelle auf Hierarchieebene j , die den gleichen Teil des Systems abbilden wie das Teilmodell b auf Ebene h
B	Menge der Teilmodelle
b	Teilmodell
$I(v)$	Menge der Variablen einer Komponente
\mathcal{V}	Verhaltensabweichung
$\mathcal{V}_Z(h)$	Zielvorgabe über die einzuhaltende Verhaltensabweichung auf Hierarchieebene h
$\omega(k)$	Gewichtung der Verknüpfung k
Θ	Menge der verwendeten Vereinfachungsoperationen

θ	Vereinfachungsoperation
$k = (v, w)$	Verknüpfung von Komponente v und w
A	Verfügbarkeit
K	Menge der Verknüpfungen
WIP	Work in Process
t	Zeit
ψ	Zustandsabbildungsfunktion
$s(b)$	Zustand eines Teilmodells
$s(B.)$	Zustand einer Menge von Teilmodellen
S	Menge der Zustände
$Z_{blc}(v)$	Anzahl der Zyklen im Block in dem v liegt
z_v	Anzahl Zyklen, in denen Komponente v auftritt
Z	Anzahl der Zyklen im Graph

1 Einleitung

und da geht was du weißt
und hier kommt was du ahnst
und das wie es wirklich ist
gegen dort wo du mal warst

(Anders als Gedacht, Kettcar)

1.1 Motivation zum Einsatz vereinfachter Modelle in der Materialflusssimulation

Der Lebenszyklus von Produkten wird immer kürzer, zur gleichen Zeit nehmen die Entwicklungskosten immer stärker zu. Um in dieser Situation auf dem Weltmarkt konkurrenzfähige Produkte anbieten zu können, müssen neue Produkte schnell und in guter Qualität in Serie gehen. Zudem muss die geplante Produktionsmenge nach Serienanlauf schnell erreicht werden. Die Simulation bietet im Rahmen der Vision „Digitale Fabrik“ eine Möglichkeit, die Planungstätigkeiten zur Erreichung dieser Ziele zu unterstützen und zu verifizieren (Digitale Planungsabsicherung). Die Materialflusssimulation, als ein Teilgebiet der Simulation, wird im Bereich der Produktionsplanung immer stärker eingesetzt. Ziele dieser Planung sind die Optimierung der logistischen Abläufe, die Layoutplanung und die Kapazitäts- und Bestandsplanung.

Durch die steigende Einsatzhäufigkeit, Nachweis der Wirksamkeit und die besser werdende Bedienbarkeit steigt der Wunsch, immer umfangreichere Simulationsmodelle zu erstellen. Am besten soll das gesamte Unternehmen mit allen Abläufen in einem Modell zusammengefasst werden. Gepaart mit dem Wunsch, immer realistischere Ergebnisse und Darstellungen zu bekommen, entstehen immer detailliertere und größere Modelle. Diese sind auch auf den immer leistungsfähiger werdenden Computern nicht mit der für effizientes Experimentieren erforderlichen Geschwindigkeit berechenbar.

Mueck [Mue05] hat in seiner Dissertation ein System entwickelt, welches die Laufzeit komplexer Simulationsmodelle diesen Zielen anpasst. Das Problem wird gelöst, indem Teilmodelle geringerer Komplexität verwendet werden, wenn diese nicht im Fokus der Analyse stehen oder keine starken Auswirkungen auf die aktuelle Analyse haben. Kernpunkt dieses Ansatzes ist, dass mit sinkender Komplexität eines Modells auch der Rechenbedarf während der Simulation sinkt. Dieser Ansatz ist nicht neu, doch mussten vor der Arbeit von Mueck Teilmodelle per Hand ausgetauscht, d.h. durch Modelle geeigneter Komplexität ersetzt werden. Zudem konnte dieser Austausch nur vor Simulationsbeginn erfolgen. Jetzt ist es möglich, dass die Komplexität der Simulation sich während der Laufzeit dem Analysefokus des Benutzers anpasst.

Ein Nachteil der bis jetzt vorliegenden Arbeiten als auch der Modellvereinfachung im Allgemeinen ist, dass die Komplexitätsreduzierung eine manuelle Tätigkeit für Simulationsexperten ist. Es müssen für einen Systembereich (z.B. Rohbau, Montageinsel im Rohbau) mehrere Teilmodelle unterschiedlicher Komplexität erstellt und gepflegt werden. Insbesondere die Pflege der über lange Zeit dynamisch weiterentwickelten Modelle ist sehr aufwendig. So zieht der Vorteil der detaillierungsvarianten Simulation den Nachteil eines erhöhten Modellierungsaufwands nach sich.

1.2 Ziele und Vorgehensweise

Ziel dieser Arbeit ist es, den Nachteil des erhöhten Modellierungsaufwands zu beseitigen, welcher durch die Nutzung von Modellen mit lokal variierender Komplexität entsteht. Es ist ein System zu entwickeln, bei dem der Modellierer nur das Modell mit höchster Komplexität erstellt und vereinfachte Teilmodelle automatisch erzeugt werden.

Im Rahmen dieser Dissertation wird folgendes Vorgehen gewählt: Das Ausgangsmodell auf der höchsten Komplexitätsebene wird zuerst iterativ partitioniert, so dass sich eine Hierarchie von immer umfassenderen Partitionen bildet. In den ersten Schritten der Iteration werden Bereiche in einer Partition zusammengefasst, die kleine, örtlich und logistisch zusammenhängende Einheiten bilden, bspw. einzelne Fertigungsinseln. In den nächsten Iterationsschritten werden die Partitionen größer (Werkstätten, Fertigungshallen,...). Dann werden für jede Partition verhaltensähnliche Teilmodelle geringerer Komplexität erstellt. Die Komplexität eines Teilmodells soll dabei kleiner sein als die Komplexität aller Teilmodelle, die es in der Hierarchie überdeckt. Es entsteht also eine Hierarchie bezüglich der

Komplexität, auf jeder höheren Hierarchiestufe nimmt die Gesamtkomplexität des Modells ab. Da eine Vereinfachung eine Verhaltensveränderung der erzeugten Teilmodelle im Verhältnis zum Ausgangsmodell induziert, nimmt auch die Verhaltensähnlichkeit des Gesamtmodells mit jeder Hierarchiestufe ab. Da eine definierte Hierarchie erstellt werden soll, werden die Komplexität und die Verhaltensabweichung als Regelgrößen der Vereinfachung verwendet, um Teilmodelle mit definierten Eigenschaften zu erzeugen. Um eine Modellrekomposition während der Laufzeit der Simulation durchführen zu können, bei der Teilmodelle ausgetauscht werden, werden in die Hierarchie der Teilmodelle Abbildungsfunktionen integriert, die aus dem Zustand der entnommenen Teilmodelle einen Zustand für die eingefügten Teilmodelle berechnet, der dem Zustand der entnommenen Teilmodelle ähnlich ist.

Diese Arbeit ist wie folgt gegliedert: In Kapitel 2 werden das Problem der geregelten Vereinfachung hierarchischer Partitionen von Modellen in der Materialflusssimulation definiert und die Anforderungen, die an die zur Lösung erforderlichen Methoden gestellt werden müssen, dargestellt. In Kapitel 3 wird der Stand der Technik in den relevanten Bereichen bezüglich der gestellten Anforderungen beleuchtet. Aus den Ergebnissen dieser Untersuchung wird in Kapitel 4 dargestellt, welche vorhandenen Methoden genutzt werden können und in welchen Bereichen Methoden angepasst oder neu entwickelt werden müssen, um die Anforderungen zu erfüllen. Die anzupassenden oder neu zu entwickelnden Methoden werden daraufhin in Kapitel 5 konzipiert. In Kapitel 6 wird das erstellte System durch empirische Untersuchungen auf seine Qualität und Einsetzbarkeit hin untersucht und bewertet. Eine Zusammenfassung der Arbeiten erfolgt zum Schluss in Kapitel 7.

2 Problemstellung

Flying is learning how to throw
yourself at the ground and miss.

*(The Hitchhikers Guide to the
Galaxy, Douglas Adams)*

Die Aufgabe dieses Kapitels ist es, die Problemstellung und die Anforderungen an die Problemlösung der geregelten Vereinfachung hierarchischer Partitionen von Modellen in der Materialflusssimulation präzise darzulegen. Dazu soll zuerst die Problemstellung dieser Arbeit in Teilprobleme zerlegt werden und diese mitsamt ihren Zielen und Randbedingungen detailliert beschrieben werden. Darauf folgend sollen die Anforderungen an Methoden zur Lösung der definierten Teilprobleme dargestellt werden.

2.1 Problemdefinition

Neben der detaillierten Beschreibung der durch Zerlegung des Gesamtproblems gebildeten Teilprobleme soll die begriffliche Basis für diese Arbeit gelegt werden, indem wichtige Begriffe definiert werden. Zunächst wird in das Themenfeld und die Begrifflichkeit der Modellierung und der Simulation im Bereich der Materialflusssimulation eingeführt. Daraufhin wird das Teilproblem der Vereinfachung von Modellen in der Materialflusssimulation dargestellt. Diese relativ allgemeinen Teilprobleme werden gefolgt von speziellen Problemen, die aus der Anwendung der dynamisch detaillierenden Simulation von Mueck [Mue05] folgen. Nachdem diese im dritten Abschnitt detailliert beschrieben ist, werden die Teilprobleme Partitionierung und Hierarchisierung von Modellen, Regelung der Vereinfachung, Generierung von konsistenten Zuständen und Finden einer geeigneten Plattform für die Implementierung dargestellt.

2.1.1 Modelle in der Materialflusssimulation

In dieser Arbeit soll mit Materialflussmodellen gearbeitet werden.

Definition 1 (Materialfluss). *Materialfluss ist die Verkettung aller Vorgänge beim Gewinnen, Be- und Verarbeiten, sowie der Verteilung von stofflichen Gütern innerhalb festgelegter Bereiche. Zum Materialfluss gehören alle Vorgänge während des Durchlaufs von Gütern (z.B. Material, Stoffmengen, Abfall, Datenträger usw.) durch ein System, wie Bearbeiten, Handhaben, Transportieren, Prüfen, Aufenthalte und Lagerungen. ...[VDI73]*

Der Materialfluss in einem System wird für eine Simulation mittels eines Modells abgebildet.

Definition 2 (System). *Nach Ordnungsprinzipien gegliederte Mannigfaltigkeit von materiellen Dingen, Prozessen usw. (materielles System) oder von Begriffen, Aussagen usw. (ideelles System). ... [KB87].*

Definition 3 (Modell). *Ein Objekt, das auf der Grundlage einer Struktur-, Funktions- oder Verhaltensanalogie zu einem entsprechenden Original von einem Subjekt eingesetzt und genutzt wird, um eine bestimmte Aufgabe lösen zu können, deren Durchführung mittels direkter Operation zunächst oder überhaupt nicht möglich bzw. unter gegebenen Bedingungen zu aufwendig ist. ... [KB87]*

Die Materialflusssimulation ist ein Spezialgebiet der Simulation.

Definition 4 (Simulation). *Simulation ist das Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierfähigen Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind [VDI00].*

Die computerbasierte Simulation kann in drei Bereiche aufgeteilt werden: die kontinuierliche, die ereignisdiskrete und die hybride. Bei der kontinuierlichen Simulation ändert sich der Zustand der Simulation entlang einer kontinuierlichen Zeitleiste, bei der hybriden sind Elemente der kontinuierlichen und diskreten kombiniert.

Definition 5 (Diskrete Simulation). *Die diskrete Simulation beschäftigt sich mit der Nachbildung von sich in der Zeit entwickelnden Systemen durch eine Repräsentation, in der sich der Simulationszustand sofortig an bestimmten, abzählbaren Zeitpunkten ändert [LK00].*

In der ereignisdiskreten Simulation treten zu Zeitpunkten Ereignisse auf, an denen sich der Simulationszustand ändert.

Definition 6 (Ereignis). *Atomare Begebenheit, die eine Änderung des Simulationszustandes bewirkt und keine Zeit verbraucht [VDI96].*

Die Materialflusssimulation wird in dieser Arbeit als eine ausschließlich ereignisdiskrete Simulation angesehen. In der computerbasierten ereignisdiskreten Simulation wird die Zustandstransformation, die beim Eintritt eines Ereignisses vollzogen werden soll, mittels einer Ereignisroutine modelliert. Tritt ein Ereignis ein, so wird die zugehörige Ereignisroutine ausgeführt.

Definition 7 (Ereignisroutine). *Einem Ereignis oder einem Ereignistyp zugeordnetes Unterprogramm, das den Simulationszustand in einer festgelegten Weise ändert, wenn das zugehörige Ereignis eintritt [LK00].*

Die Beschreibung, die Notation, von Modellen der Materialflusssimulation kann auf vielfältige Weise geschehen [MSJ93]. Allen heutzutage Bedeutenden ist gemein, dass sie einen objektorientierten Ansatz verfolgen und somit Klassen und Instanzen von Simulationsobjekten, im Weiteren Komponenten genannt, verwalten. Modelle bestehen aus verknüpften, von Komponentenklassen abgeleiteten Komponenten. Das im Modell befindliche Material wird durch Materialobjekte, im Weiteren Marken genannt, abgebildet, die zwischen den Komponenten über Verknüpfungen verschickt und verändert werden können. Da Modelle oftmals zur besseren Analyse visuell in einer 2D oder 3D Darstellung angezeigt werden, erhalten Komponenten hierzu Zusatzinformationen: einen visuellen Repräsentanten und eine Position im Raum. So entsteht eine logische und geometrische Abbildung des Systems.

2.1.2 Vereinfachung von Materialflussmodellen

Durch die Vereinfachung eines Modells, des Ausgangsmodells, soll ein neues Modell erzeugt werden, welches dasselbe System abbildet, aber dessen Simulation eine bessere Laufzeit besitzt.

Definition 8 (Ausgangsmodell). *Das Modell, welches a priori vorhanden ist und vereinfacht werden soll.*

Definition 9 (Laufzeit von Materialflusssimulationen). *Die Laufzeit einer Simulation ist die Menge Realzeit, die für die Berechnung einer Menge Simulationszeit¹ benötigt wird. Die Laufzeit einer Simulation - abstrahiert vom verwendeten Rechner und der Software - wird bestimmt durch die Anzahl auftretender Ereignisse bezogen auf eine Menge Simulationszeit und der vom Rechner benötigten Realzeit zur Berechnung der zugehörigen Ereignisroutinen (vgl. [HL99]).*

Existieren zwei unterschiedliche Modelle, die dasselbe System abbilden, dann soll das Modell mit der schlechteren Laufzeit als komplexer als das andere Modell bezeichnet werden. Im Umkehrschluss soll gelten, dass von zwei Modellen, die dasselbe System abbilden, das mit der höheren Komplexität eine längere Laufzeit besitzt.

Definition 10 (Komplexität von Materialflussmodellen). *Die Komplexität wird bestimmt durch die Anzahl der Komponenten des Modells, der zwischen den Komponenten bestehenden logischen Verknüpfungen und den Ereignisroutinen² [SY93].*

Wesentlich wird die Anzahl auftretender Ereignisse bestimmt durch die von außen angelegte Systemlast, dem Produktionsprogramm, der Anzahl zu bearbeitender Aufträge. In der Simulation eines Modells, egal aus wie vielen Komponenten es besteht, werden keine Ereignisse auftreten, wenn keine Systemlast angelegt ist. Im Rahmen der Vereinfachung eines Modells ist die Systemlast der Vereinfachung gleich der des Ausgangsmodells. In einem Modell bestehend aus einer größeren Menge Komponenten muss ein Auftrag auf seinem Weg durch das Modell mehr Komponenten durchlaufen als in einem Modell bestehend aus einer kleineren Menge Komponenten. Sind die Komponenten eines Modells aufwendig verknüpft, so müssen häufig Entscheidungen über die Route der Aufträge durch das System getroffen werden. Dies führt ebenfalls zu einer erhöhten Anzahl auftretender Ereignisse.

Nachfolgend zu der Definition von Komplexität, die direkt aus der Laufzeit der Simulation hergeleitet wurde, kann die Vereinfachung von Modellen definiert werden als:

Definition 11 (Vereinfachung von Materialflussmodellen). *Eine Vereinfachung ist die Erzeugung eines neuen Modells, welches dasselbe System abbildet wie das Ausgangsmodell, aber eine geringere Komplexität besitzt.*

¹Die Simulationszeit ist eine Modellgröße, die die im realen System voranschreitende Zeit im Modell abbildet und die durch einen Zeitfortschaltungsmechanismus des Simulators erhöht wird [VDI96].

²Die Faktoren Anzahl und Verknüpfung der Komponenten werden von mehreren Autoren genannt [BT00, KB87, Wal87, Völ03]. Um eine bessere Anpassung an die Laufzeit zu erreichen, wurde als weiterer Faktor die Ereignisroutinen hinzugenommen [SY93].

Ein durch Vereinfachung erzeugtes Modell weist einen Unterschied im Aufbau im Vergleich zum Ausgangsmodell auf, da es aus weniger und anders verknüpften Komponenten besteht. Bei einer schwachen Vereinfachung wird nur ein leicht verändertes Modell erzeugt, bei einer großen Vereinfachung ein stark verändertes Modell. Durch den Unterschied im Aufbau entsteht während einer Simulation ein Unterschied in dem Verhalten des vereinfachten Modells im Vergleich zum Ausgangsmodell. Mit der Stärke der Vereinfachung nimmt die erzeugte Verhaltensabweichung zu, weil der Unterschied im Aufbau der Modelle wächst.

Definition 12 (Verhaltensabweichung). *Die Verhaltensabweichung ist der Unterschied, der Fehler, im Verhalten der Simulation - der Änderung des Simulationszustands über der Simulationszeit - des durch Vereinfachung erzeugten Teilmodells in Bezug zum Ausgangsmodell.*

Eine zulässige Vereinfachung liegt dann vor, wenn das erzeugte Modell valide ist. Dies ist der Fall, wenn die Verhaltensabweichung einen vorgegebenen, vom Untersuchungszweck abhängigen Schwellenwert nicht übersteigt.

Definition 13 (Validität). *Ein Modell ist valide, wenn es eine akkurate Repräsentation für einen bestimmten Untersuchungszweck darstellt [LK00].*

2.1.3 Die detaillierungsvariante diskrete Simulation

Mueck [Mue05] entwickelte das Verfahren der detaillierungsvarianten diskreten Simulation³.

Definition 14 (Detaillierungsvariante diskrete Simulation (dvdS)). *Um die Laufzeit einer Simulation zu verbessern, kann die Komplexität eines Modells lokal variiert werden. Bereiche des Systems, die in einem speziellen Experiment nicht im Fokus des Untersuchungsinteresses stehen, werden durch Teilmodelle geringerer Komplexität abgebildet. Dazu erfolgt eine Komposition von Teilmodellen zu einem das gesamte System abbildenden dvdS-Modell während der Laufzeit. Ändert sich der Fokus des Untersuchungsinteresses während der Simulation, erfolgt eine Rekombination des dvdS-Modells. Durch in die Hierarchie integrierte Zustandsabbildungsfunktionen wird der Zustand der Simulation über Rekombinationen hinweg konsistent gehalten.*

³Der von Mueck geprägte Begriff der detaillierungsvarianten Simulation (dvdS) wird im Folgenden beibehalten, auch wenn nicht die Detaillierung, sondern die Komplexität betrachtet wird.

Definition 15 (Teilmodell). *Ein Teilmodell ist ein Modell, das einen Teil des Systems abbildet, gebildet aus einer Partition des Ausgangsmodells oder der Vereinfachung anderer Teilmodelle.*

Die (Re-)Komposition von dvdS-Modellen soll aus mehrstufig komplexitätsreduzierten Teilmodellen erfolgen, um eine feine Anpassung von Komplexität und Verhaltensabweichung auf den Untersuchungsfokus vornehmen zu können. Zur Verwaltung der Teilmodelle soll eine Hierarchie Verwendung finden.

Die unterste Ebene dieser Hierarchie besteht aus Teilmodellen, die ohne Vereinfachung aus den Partitionen des Ausgangsmodells erzeugt wurden, die in der ersten Iterationschleife der Partitionierung entstanden (siehe Abbildung 2.1). Die nächsthöhere Ebene besteht aus Teilmodellen, gebildet durch die Vereinfachung von mehreren Teilmodellen der nächstniedrigeren Ebene. Welche Teilmodelle dazu verwendet werden, wird durch die entsprechende Partition der iterativen Partitionierung festgelegt. Der pro Teilmodell abgebildete Bereich des Systems wächst mit der Höhe der Ebene in der Hierarchie. Dadurch wird erreicht, dass auf jeder Hierarchieebene das gesamte System abgebildet wird und die Anzahl und Komplexität der Teilmodelle beständig sinkt. Jedes Teilmodell besitzt in der Hierarchie ein zugeordnetes Teilmodell niedrigerer Komplexität und/oder mehrere Teilmodelle höherer Komplexität. Die Anzahl der Hierarchieebenen ist nicht vorgegeben. In der dvdS wird immer das gesamte - ohne Überschneidungen und Leerstellen - System simuliert, darum ist nur eine Auswahl an Teilmodellen aktiv und verändert den Simulationszustand. Die restlichen inaktiven Teilmodelle werden nicht simuliert, dort treten somit keine Ereignisse auf. Die Entscheidung, welche Teilmodelle aktiv oder inaktiv sein müssen, um eine korrekte dvdS zu erlauben und die Komplexität und Verhaltensabweichung für bestimmte Bereiche des Systems festzulegen, wird mit Hilfe der Hierarchie getroffen.

Während der Durchführung der dvdS, also nach der Bildung der Hierarchie und somit der Erzeugung aller Teilmodelle, kann sich der Untersuchungsfokus ändern. Dann soll eine Rekombination der Teilmodelle erfolgen. Aktive Teilmodelle werden deaktiviert, inaktive aktiviert (siehe Abbildung 2.2). Die deaktivierten Teilmodelle bilden denselben Teil des Systems ab wie die aktivierten. Daraus folgt, dass die aktivierten einen Zustand annehmen können sollten, der dem der deaktivierten ähnlich ist. Ein solcher Zustand soll mittels Zustandsabbildungsfunktionen, die in die Hierarchie integriert sind, generiert werden. So ist eine Rekombination möglich, ohne den Simulationszustand des dvdS wesentlich zu verändern, d.h. es können weiterhin brauchbare Untersuchungsergebnisse erzielt werden. Zur

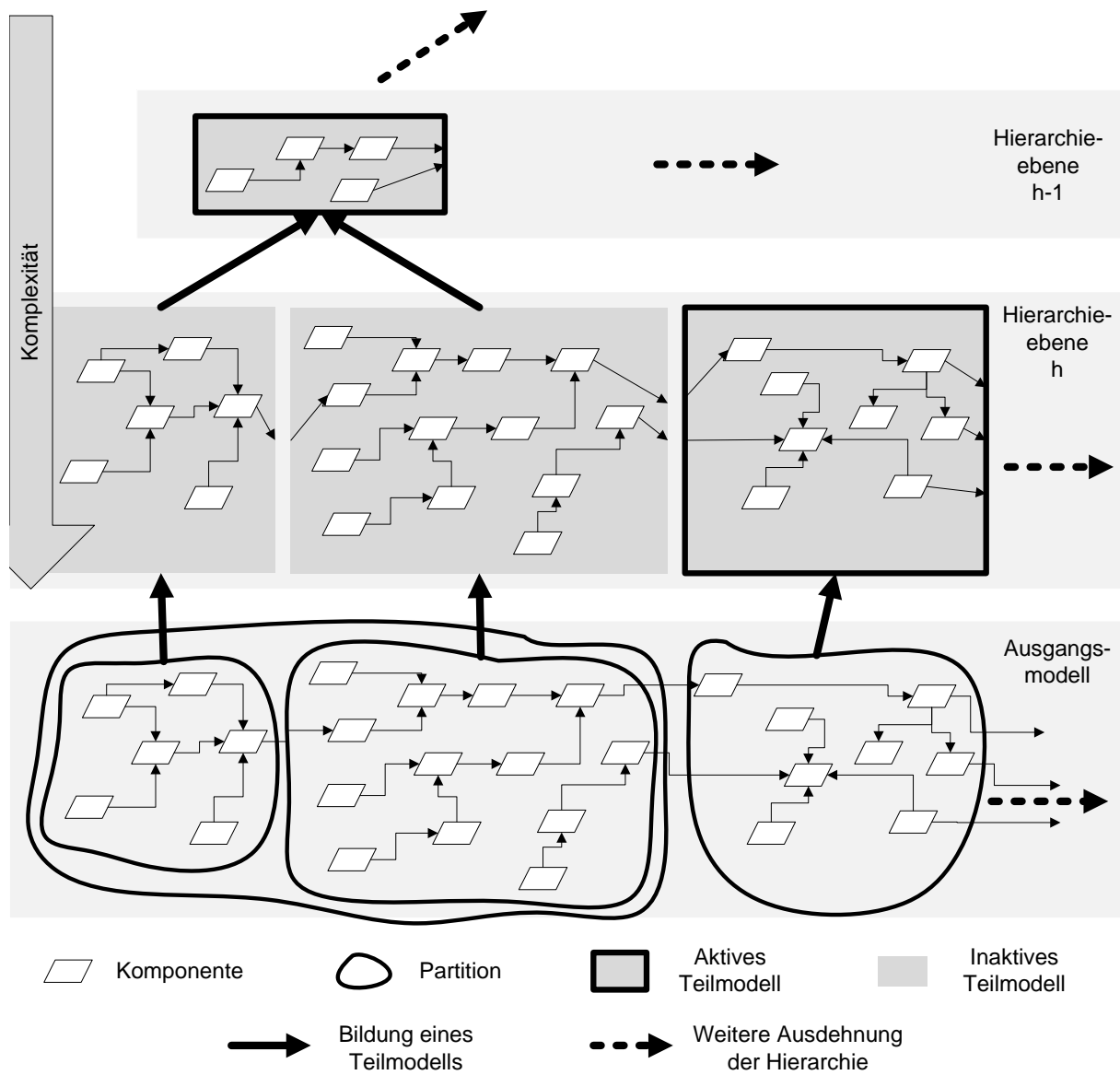


Abbildung 2.1: Aufbau eines dvdS-Modells

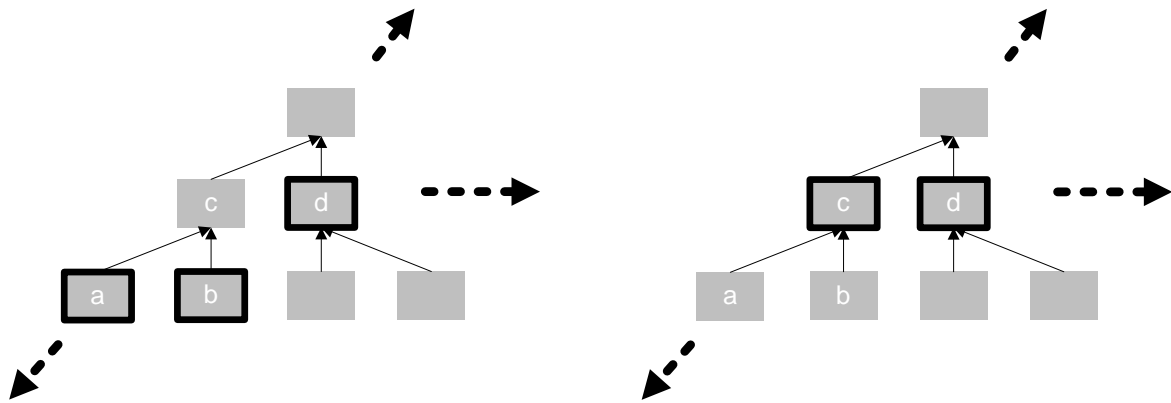


Abbildung 2.2: Rekomposition eines dvdS-Modells

Bestimmung der Relevanz eines Systembereichs für einen speziellen Untersuchungsfokus werden verschiedene Indikatoren eingesetzt. Einige benutzen die Position und Blickrichtung eines in einer interaktiven virtuellen Umgebung befindlichen Beobachters der Simulation und somit auch die Positionen und die visuellen Repräsentanten der Komponenten. Neben diesen, auf der Geometrie des Modells basierenden Indikatoren wird der logistische Zusammenhang der Komponente benutzt (siehe Abschnitt 3.6.1). Die Indikatoren sorgen dafür, dass logistisch und geometrisch an den Untersuchungsfokus angrenzende Systembereiche durch Teilmodelle bestimmter Komplexität abgebildet werden.

Das Verhalten eines Modells ist, neben dem strukturellen Aufbau, von der Parametrierung (Bearbeitungszeiten, Puffergrößen, etc.) und vom Produktionsprogramm (zu bearbeitende Aufträge, Systemlast) abhängig. Der Aufbau und die Parametrierung des Ausgangsmodells sind einzelnen Komponenten zugeordnete Eigenschaften und werden bei der Erzeugung von Teilmodellen berücksichtigt. Das Produktionsprogramm wird als Last dem System von außen auferlegt, ist keiner Komponente direkt zugeordnet und wird als in der Vereinfachung nicht erfass- oder veränderbar angesehen.

Werden Simulationsexperimente durchgeführt, dann wird sehr oft die Parametrierung geändert - hier insbesondere die Wahrscheinlichkeitsfunktionen -, seltener auch das Produktionsprogramm und der Aufbau. Durch die Eigenschaften und die Zielsetzung der dvdS ist es aber nicht nötig, dass bei jeder Variation eine vollständige Neuerzeugung der gesamten Hierarchie durchzuführen ist. Es wird angenommen, dass die häufigen Änderungen nur im Bereich des Untersuchungsfokus erfolgen. Dieser Fokus wird in der dvdS in hoher Komplexität berechnet, d.h. es wird als Teilmodell nur ein um Zusatzmetho-

den erweitertes Ausgangsmodell simuliert. Die vereinfachten Teilmodelle dieses Bereiches werden nicht aktiviert und brauchen daher nicht neu erzeugt zu werden. Die Erzeugung von wenigen Teilmodellen auf der untersten Hierarchieebene ist nicht aufwendig, da nur die für die dvdS nötigen Zusatzinformationen neu erstellt werden müssen. Durch diese Eigenschaft kann der Gesamtrechenbedarf (Vereinfachung, inkl. Regelung, und Experiment) begrenzt werden. Werden schwerwiegende Änderungen vorgenommen, so müssen betroffene Teilmodelle durch Vereinfachung neu erzeugt werden.

Aus der Nutzung der dvdS entstehen besondere Probleme, die über eine „einfache“ Modellvereinfachung hinausgehen. Dies sind die Auswahl der zu einem Teilmodell zu vereinfachenden Bereiche des Systems, die Verwaltung der Teilmodelle, die Erzeugung mehrfach in ihrer Komplexität abgestufter Teilmodelle und die Behandlung des Simulationszustands nach Rekomposition. Diese Probleme sollen in den nächsten Abschnitten detailliert beschrieben werden.

2.1.4 Partitionierung und Hierarchisierung

Die für die dvdS nötige Hierarchie von Teilmodellen soll mittels einer iterativen Partitionierung gebildet werden. Die Komponenten des Ausgangsmodells werden in relativ kleinen, logistisch und geometrisch zusammenhängenden Partitionen zusammengefasst. Die so entstandenen Partitionen werden iterativ weiter zusammengefasst. In jeder Iterationsschleife wird die Anzahl der Partitionen geringer und somit der von ihnen jeweils abgebildete Bereich des Systems größer. Jede Iterationsschleife der Partitionierung definiert die Teilmodelle einer Hierarchieebene.

Die Partitionierungsmethode hat somit einen starken Einfluss auf die Vereinfachung, da sie festlegt, welche Komponenten und ihre logistische Verknüpfung zu einem neuen Teilmodell vereinfacht werden sollen. Da in der dvdS zur Bestimmung der Relevanz eines Systembereichs auch geometrische Daten des Modelllayouts verwendet werden, muss neben dem für die Vereinfachung wichtigen logistischen auch ein geometrischer Zusammenhang der in einer Partition zusammengefassten Komponenten vorliegen.

Das klassische Partitionierungsproblem ist ein Min-Cut-Problem, bei dem versucht wird, die Menge der geschnittenen Kanten zu minimieren. Eine Partition tendiert allerdings dazu, eine umso kleinere Schnittgröße zu haben, je unbalancierter sie ist. Der Extremfall besteht, wenn alle Knoten einer einzigen Partition zugeteilt werden und die Schnittgröße

damit komplett verschwindet. Um ein solches Verhalten zu verhindern, beinhalten die meisten Methoden der Graphpartitionierung explizit oder implizit eine Restriktion, die darauf abzielt, dass alle Partitionen gleich groß sind oder in einem bestimmten Größenverhältnis zueinander stehen. Diese Restriktion darf allerdings, um nicht nur den logistischen, sondern auch den für diese Arbeit wichtigen geometrischen Zusammenhang der Komponenten in einer Partition zu gewährleisten, keine straffe Gleichgewichtsbedingung sein, sondern eine abgeschwächte. Die Einführung einer wie auch immer gearteten Gleichgewichtsbedingung bedeutet aber auch, dass sich die Partitionierung nicht wie das Min-Cut-Problem in Polynomialzeit lösen lässt, sondern NP-Vollständig ist. Die Verwendung einer Heuristik ist somit nötig.

Für die Auswahl einer geeigneten Partitionierungsmethode ist der durchschnittliche Knotenrang, also die Anzahl Vorgänger und Nachfolger einer Komponente im Modell, eine entscheidende Größe [Saa97]. In den in dieser Arbeit untersuchten Modellen, die hauptsächlich Fließfertigungen abbilden, liegt der Knotenrang im Bereich von 1 - 5 mit einem deutlichen Modus von 2. Die zu wählende Partitionierungsmethode sollte somit auf solchen schwach besetzten Graphen besonders gut abschneiden.

2.1.5 Regelung der Vereinfachung

Es wurden zwei Modelleigenschaften definiert, die bei der Vereinfachung gegenläufig beeinflusst werden, die Komplexität und die Verhaltensabweichung. Teilmodelle sollen so erstellt werden, dass sie auf jeder Hierarchieebene ungefähr dieselbe Komplexität im Verhältnis zu ihren Partitionen des Ausgangsmodells haben, damit eine Auswahl der zu aktivierenden Teilmodelle zur Einstellung der Laufzeit der dvdS erfolgen kann. Ebenfalls sollen die Teilmodelle einer Hierarchieebene ungefähr dieselbe Verhaltensabweichung aufweisen, damit bei der Auswahl der zu aktivierenden Teilmodelle auch die Validität der dvdS erhalten bleibt. Für jedes Teilmodell gibt es also Zielwerte für die zu erreichende Komplexität und Verhaltensabweichung. Damit dies gelingen kann, muss die Vereinfachung bezüglich der erreichten Komplexität und der Verhaltensabweichung steuerbar sein. Mittels der Einbettung der Vereinfachung in eine Regelung sollen die Zielwerte erreicht werden.

Definition 16 (Regelung). *Das Regeln, die Regelung, ist ein Vorgang, bei dem fortlaufend eine Größe, die Regelgröße (zu regelnde Größe), erfasst, mit einer anderen Größe,*

der Führungsgröße, verglichen und im Sinne einer Angleichung an die Führungsgröße beeinflusst wird (DIN19226).

Die primäre Regelgröße soll die Komplexität sein, ihre Führungsgröße die Zielkomplexität. Die sekundäre Regelgröße ist die Verhaltensabweichung, ihre Führungsgröße die Zielverhaltensabweichung. Ein Modell wird vereinfacht, bis die Zielkomplexität erreicht ist. Ist dies der Fall, so soll die Abweichung der Verhaltensabweichung beseitigt werden. Die Komplexität und Verhaltensabweichung müssen während der Vereinfachung mehrfach gemessen werden und die festgestellten Abweichungen zu den Führungsgrößen in einer Rückkopplungsschleife in die weitere Vereinfachung einfließen.

Definition 17 (Geregelte Vereinfachung). *In eine Rückkopplungsschleife eingebundene Vereinfachung, mit den Führungsgrößen Zielkomplexität und -verhaltensabweichung und den zu regelnden Größen der Teilmodelle Komplexität und Verhaltensabweichung.*

2.1.6 Konsistenz und Qualität des Simulationszustands nach Rekombosition

Nach Law und Kelton [LK00] ist der Simulationszustand zu einem Zeitpunkt definiert als die Menge an Zustandsvariablen, die nötig sind, um das simulierte System zu diesem Zeitpunkt zu beschreiben. Dies ist aber für die in dieser Arbeit nötigen Methoden nicht ausreichend. Deshalb soll, wie in der parallelen und verteilten Simulation [Fuj98], zusätzlich zu den Zustandsvariablen noch die Liste aller eingeplanten und noch nicht ausgeführten Ereignisse in den Zustand aufgenommen werden.

Definition 18 (Simulationszustand). *Der Zustand einer dvdS ist der Zustand aller aktivierten Teilmodelle. Der Zustand eines Teilmodells zu einem Zeitpunkt wird bestimmt durch die Werte der Zustandsvariablen, der Markenbelegung und der Liste der eingeplanten Ereignisse aller in diesem Teilmodell befindlichen Komponenten.*

Findet eine Rekombosition des Modells statt, werden inaktive Teilmodelle aktiviert. Diese Teilmodelle haben in dem Zeitraum ihrer Inaktivität ihren Zustand nicht verändert, im Gegensatz zu den bis zur Rekombosition aktiven Teilmodellen, deren Stelle sie nun einnehmen sollen. Um eine funktionsfähige Simulation zu erhalten, müssen für die zu aktivierenden Teilmodelle konsistente Zustände generiert werden. In einem konsistenten

Zustand sind keine Marken verloren gegangen und die Simulation enthält keine Komponenten, die Zustände haben, die sie nicht selbst hätten herbeiführen können. Zusätzlich soll sich der Zustand der Gesamtsimulation nach der Rekombination möglichst wenig von dem unterscheiden, den sie ohne Rekombination gezeigt hätte. Bestimmen die durch die Zustandsabbildung generierte Belegung und die erzeugten eingeplanten Ereignisse der zu aktivierenden Teilmodelle in weiten Teilen das kurzfristige weitere Verhalten, so wird der Zustand zusätzlich durch die Zustandsvariablen (bspw. statistische Daten) definiert.

Mueck [Mue05] entwickelte zur Zustandsabbildung mehrere unterschiedliche Methoden. Eine dieser Methoden sieht vor, dass jedes Teilmodell eine Zustandsabbildungsfunktion beinhaltet, die angibt, wie der Zustand des Teilmodells aus dem oder den Zuständen der über die Hierarchie zugeordneten Teilmodelle errechnet werden kann. Diese Abbildungsfunktionen müssen allerdings vom Modellierer implementiert werden. Um ein komplett automatisiertes Verfahren zu schaffen, sollen diese Abbildungsfunktionen automatisch im Rahmen der geregelten Vereinfachung erstellt werden.

Da über die Hierarchie jedem vereinfachten Teilmodell mehrere komplexere Teilmodelle zugeordnet sind, werden bei einer Rekombination entweder mehrere Teilmodelle aktiviert oder mehrere deaktiviert. Der Zustand von mehreren Teilmodellen muss demnach auf eines übertragen werden oder der Zustand eines Teilmodells muss auf mehrere übertragen werden. Um eine Kapselung von Modellen zu erlauben, sollen die Zustandsabbildungsfunktionen nicht wie bei Mueck in die Teilmodelle integriert werden, sondern in die Hierarchie, die Zugriff auf alle Teilmodelle hat.

Durch eine Zustandsabbildungsfunktion, die einen geringeren Aufwand hat als eine Simulation des zu aktivierenden Teilmodells, kann aus offensichtlichen Gründen⁴ kein Zustand für das zu aktivierende Teilmodell generiert werden, der dem des zu deaktivierenden entspricht. Auch kann das Verhalten des aktivierten Modells nicht dem des deaktivierten entsprechen, da es sich um unterschiedliche Modelle handelt. Es kann demnach nur eine möglichst große Ähnlichkeit in Zustand und Verhalten erreicht werden.

⁴Andernfalls würde es bedeuten, dass die Simulation als Methode zur Analyse komplexer Systeme komplett sinnlos wäre, weil Ergebnisse schneller mit anderen Mitteln erlangt werden könnten.

2.1.7 Integration des Verfahrens in das Simulationswerkzeug d³FACT

Zur Validierung der entwickelten Methoden soll eine Implementierung der geregelten Vereinfachung hierarchischer Partitionen im Framework des Simulationswerkzeugs d³FACT erfolgen. d³FACT ist ein Werkzeug zur diskreten Simulation und Visualisierung von Materialflussmodellen, das von Laroque [Lar07] entwickelt wurde. Das Verfahren zur dvdS wurde bereits teilweise in dieses Werkzeug integriert. Der Simulator ist ein in Entwicklung befindliches, offenes Softwareprojekt. Alle Komponenten des Simulatorframeworks können in dieser Arbeit weitreichend verändert werden, so dass wenige Einschränkungen bzgl. des Simulators in der Implementierung und Konzeption entstehen. Die Implementierung im Framework von d³FACT bringt mehrere Vorteile gegenüber den Alternativen. Kommerzielle Simulatoren sind im Allgemeinen Closed Source⁵ und können nicht verändert werden. Gegenüber anderen offenen Simulatoren bietet d³FACT den Vorteil, dass die dvdS schon teilweise implementiert ist. Eine Neuimplementierung eines Simulators wird aus Aufwandsgründen nicht in Betracht gezogen. Für diese Arbeit relevante Eigenschaften des Simulators d³FACT sind die Verwendung von Java als Programmiersprache, die Verwendung von XML als Modelldatenstruktur und die individuelle Methode der Modellbeschreibung.

2.2 Anforderungen an die Problemlösung

In diesem Abschnitt sollen die Anforderungen festgelegt werden, die an die Lösungen der Probleme aus Abschnitt 2.1 gestellt werden müssen, um das Ziel dieser Arbeit zu erfüllen. Dies sind Anforderungen an die zu wählende Modellbeschreibung, an die Partitionierung und Hierarchisierung, an die Vereinfachung, an die Regelung, an die Zustandsgenerierung und Anforderungen, die sich aus der Wahl des Frameworks für die Implementierung ergeben.

⁵Im Gegensatz zu Open Source wird bei Closed Source der Quellcode nicht preisgegeben und kann somit auch nicht verändert werden.

2.2.1 Anforderungen an die Modellbeschreibung

Die Modellbeschreibung muss mehreren Anforderungen genügen. Triviale und daher nicht näher spezifizierte Anforderungen sind, dass sich mit der Modellbeschreibung erstellte Modelle in einem diskreten Simulator ohne großen Aufwand ausführen lassen müssen und dass sie für die Modellierung von Materialflusssystemen geeignet sein muss. Zudem soll die Modellbeschreibung so gewählt werden, dass sie die Algorithmen der geregelten Vereinfachung unterstützt, also gewährleistet, dass die Komplexität und die Verhaltensabweichung einfach gemessen werden können, die Partitionierung ihren Anforderungen entsprechen und eine Vereinfachung durchgeführt werden kann.

Um eine automatisierte und geregelte Vereinfachung ermöglichen zu können, müssen die Modelllogik und die Modellstruktur durch Algorithmen analysierbar sein. Bei vielen Modellbeschreibungen gängiger Simulatoren⁶ ist dies durch die Verwendung frei programmierbarer Komponenten nicht der Fall. In diesen Komponenten kann der Modellierer unstrukturiert sehr komplexes Verhalten durch die Verwendung von werkzeugabhängigen Skriptsprachen definieren. Eine automatische Analyse der so entstehenden Ereignisroutinen ist nur sehr schwer möglich. Deshalb soll die komplexe Logik eines Systems mittels standardisierter und nur parametrierbarer Komponenten abgebildet werden können.

Anforderung 1 (Automatisierbare Modellanalyse). *Die Modellbeschreibung muss so gewählt werden, dass der Modellgraph aus Komponenten mit standardisierten Eigenschaften besteht, damit diese mittels Algorithmen analysiert werden können.*

Diese Anforderung erzeugt die Gefahr, dass nur noch sehr spezielle Systeme modelliert werden können. Dies soll vermieden werden, indem die Flexibilität der Modellierung weitestgehend erhalten bleiben soll.

Anforderung 2 (Erhalt der Flexibilität der Modellierung). *Durch die Wahl der Modellbeschreibung soll die Vielfalt der modellierbaren Systeme nicht signifikant eingeschränkt werden.*

Eine weitere Gefahr bei der Wahl einer speziellen Modellbeschreibung besteht darin, dass bspw. Eigenschaften für eine starke Vereinfachung integriert werden, die die Modelle oder

⁶Als Beispiele sollen hier eM-Plant (Produkthomepage: www.emplant.de) und der Stand von d³FACT vor dieser Arbeit genannt werden.

die Modellausführung komplexer machen als sie für eine gewöhnliche - keine dvdS - Simulation nötig wären. Es entstünde die paradoxe Situation, dass diese Modelle gut zu vereinfachen wären, es aber auch werden müssten, um den gleichen Nutzen wie gewöhnliche Modelle zu erreichen.

Anforderung 3 (Schlankheit der Modelle und Modellausführung). *In die Modellbeschreibung soll der Anteil an Eigenschaften, die einzig für die Durchführung von in dieser Arbeit verwendeten Methoden verwendet werden, so gering wie möglich gehalten werden.*

Für den Simulator d³FACT, in den die geregelte Vereinfachung integriert werden soll, wurden in den letzten Jahren einige sehr komplexe Modelle erstellt. Zudem ist sehr viel Humankapital in die Entwicklung des Simulators geflossen. Um diese Arbeit erhalten zu können, soll die Modellbeschreibung so gewählt werden, dass nur wenige Änderungen in bestehenden Modellen und dem Simulator getätigt werden müssen.

Anforderung 4 (Geringe Änderungen im Simulator d³FACT). *Da die geregelte Vereinfachung in den Simulator d³FACT integriert werden soll, ist darauf zu achten, dass möglichst wenige Änderungen zur bestehenden Modellbeschreibung vollzogen werden.*

In der dvdS findet eine (Re-)Komposition von Teilmodellen statt. Damit das Verbinden von neu aktivierten Teilmodellen mit anderen aktiven funktionieren kann, müssen Teilmodelle über definierte Schnittstellen verfügen. In der in d³FACT bestehenden Modellbeschreibung sind nur Marken als Ein- und Ausgaben von Teilmodellen erlaubt. Alle anderen zur Simulation nötigen Daten sind im Teilmodell selbst vorhanden.

Anforderung 5 (Gekapselte Teilmodelle). *Die Modellbeschreibung hat so zu erfolgen, dass alle Interaktionen mit einem Teilmodell über definierte Schnittstellen erfolgen. Bis auf ein- und ausgehende Marken müssen alle Informationen zur Simulation des Teilmodells in ihm selbst vorliegen.*

2.2.2 Anforderungen an die Partitionierung und Hierarchisierung

Die dvdS entscheidet anhand von logistischen und geometrischen Indikatoren (vgl. Abschnitt 3.6.1), welche Teilmodelle in der dvdS aktiv sind. Die Eigenschaften der Partitionen entscheiden, wie effizient die dvdS arbeitet, weil sie die Teilmodelle definieren.

Die Komponenten einer Partition sollen einen geometrischen Zusammenhang haben, damit in der dvdS eine gute Auswahl von Teilmodellen zur Aktivierung anhand der geometrischen Indikatoren erfolgen kann.

Anforderung 6 (Geometrischer Zusammenhang der Partitionen). *Zu einer Partition sollen Modellkomponenten so zusammengefasst werden, dass eine räumliche Trennung von Komponenten unterschiedlicher Partitionen vorliegt und Partitionen somit geschlossene Formen bilden.*

Wegen des logistischen Indikators sollen die Komponenten einer Partition zudem einen logistischen Zusammenhang haben, so dass in einer Partition keine Komponenten sind, die auf den Materialfluss bezogen keinen oder nur entfernten Zusammenhang besitzen. Dadurch wird erreicht, dass Teilmodelle möglichst wenig Schnittstellen besitzen, wodurch eine effektivere Vereinfachung unternommen werden kann, denn liegen viele Schnittstellen vor, so existieren mit großer Wahrscheinlichkeit viele Verzweigungen im Markenfluss, die eine Zusammenfassung von Modellkomponenten erschweren können.

Anforderung 7 (Logistischer Zusammenhang der Partitionen). *Logistisch eng zusammenhängende Modellkomponenten sollen in einer Partition zusammengefasst werden, dabei ist zu berücksichtigen, dass eine Partition möglichst wenige Verknüpfungen zu anderen Partitionen besitzen soll.*

Damit Rekompersionsvorgänge die Komplexität der dvdS effektiv anpassen können, sollen alle Partitionen auf einer Hierarchieebene in etwa denselben Umfang haben. Ist dies nicht der Fall, so werden teilweise in sehr vielen Umschaltungen sehr viele, nur kleine Bereiche des Systems abbildende Teilmodelle ausgetauscht oder teilweise solche, die weite Bereiche des Systems abbilden. Bei ersterem wird der Vorteil der dvdS, die Laufzeit der dvdS zu verbessern, durch den großen Aufwand des Aktivierens und Deaktivierens von vielen Teilmodellen zunichtegemacht. Bei letzterem kann die Komplexität der Gesamtheit der aktivierten Teilmodelle in einer dvdS nur grob angepasst werden.

Anforderung 8 (Ausgeglichener Umfang der Partitionen). *Die Partitionierung hat so zu erfolgen, dass alle Partitionen einer Hierarchieebene einen ähnlich großen Bereich des Systems abbilden.*

Eine weitere, pragmatische Anforderung ergibt sich, damit die dvdS durchführbar ist und verwertbare Ergebnisse erzeugt.

Anforderung 9 (Validität der Hierarchie). *Die Partitionierung muss so erfolgen, dass ein Baum entsteht, der mindestens zwei Ebenen besitzt. Auf jeder Hierarchieebene ist jede Komponente in genau einer Partition.*

2.2.3 Anforderungen an die automatisierte Modellvereinfachung

Zentraler Aspekt dieser Arbeit ist, dass die Komplexität und Verhaltensabweichung eines Teilmodells gemessen werden und eine angepasste Vereinfachung solange durchgeführt wird, bis die Zielvorgaben erfüllt werden oder festgestellt wird, dass die Zielvorgaben nicht erfüllt werden können. Damit dies erfolgen kann, muss der Vereinfachungsalgorithmus ein beeinflussbares, deterministisches Verhalten zeigen.

Anforderung 10 (Regelbarkeit des Vereinfachungsalgorithmus). *Der Vereinfachungsalgorithmus muss so steuerbar sein, dass die Komplexität und Verhaltensabweichung erzeugter Teilmodelle im Sinne einer Regelung den Zielvorgaben angenähert werden können.*

Es soll eine Hierarchie von Teilmodellen mit unterschiedlicher Komplexität aufgebaut werden. Die Anzahl der Hierarchieebenen ist dabei nicht vordefiniert, sondern kann je nach Modell und Untersuchungsziel „frei“ gewählt werden, wobei eine Hierarchietiefe von 3-5 als am wahrscheinlichsten für den praktischen Einsatz anzusehen ist.

Anforderung 11 (Nahezu stufenlos erreichbare Vereinfachungsgrade). *Die erreichbaren Vereinfachungsgrade sollen nahezu stufenlos einstellbar sein.*

Das zufallsabhängige Verhalten von Simulationsmodellen ist eine wesentliche Eigenschaft, die besonders wichtig für die Aussagekraft von Simulationsexperimenten ist.

Anforderung 12 (Erhalt des stochastischen Verhaltens). *Die Charakteristika der Zufallsverteilungen von Variablen der Komponenten sollen bei vereinfachten Modellen bis zu einem hohen Vereinfachungsgrad weitgehend erhalten bleiben.*

2.2.4 Anforderung an die Regelung des Vereinfachungsalgorithmus

Zwei Maße sollen die Komplexität und die Verhaltensabweichung eines durch Vereinfachung erzeugten Teilmodells messen. Die Messwerte der beiden Maße sollen als Regelgrößen der Vereinfachung dienen.

Anforderung 13 (Teilmodelle, die den Vorgaben entsprechen). *Die Regelung muss die Vereinfachung so steuern, dass erzeugte Teilmodelle den Zielwerten bezüglich Komplexität und Verhaltensabweichung entsprechen.*

Da die Zielstellung der Komplexitätsreduktion die Verbesserung - Verkürzung - der Laufzeit der Simulation ist, muss das Komplexitätsmaß Messwerte liefern, die eine Aussage über die Laufzeit der Simulation eines Modells erlauben. Es erscheint schwierig, einen funktionalen Zusammenhang zwischen dem Messergebnis \mathcal{C} und der Laufzeit in der Art $\mathcal{L} = a\mathcal{C} + b$ zu erzeugen. Das Maß soll aber einen vergleichenden Zusammenhang zwischen Komplexität und Laufzeit herstellen, der wie folgt aussieht: $\frac{\mathcal{C}(p_{i,h})}{\mathcal{C}(b_{i,h})} \approx \frac{\mathcal{L}(p_{i,h})}{\mathcal{L}(b_{i,h})}$ ⁷.

Anforderung 14 (Starke Korrelation zwischen Komplexitätsmaß und Laufzeit). *Das Komplexitätsmaß soll für Teilmodelle Kennzahlen generieren, die eine starke Korrelation zwischen $\frac{\mathcal{C}(p_{i,h})}{\mathcal{C}(b_{i,h})}$ und $\frac{\mathcal{L}(p_{i,h})}{\mathcal{L}(b_{i,h})}$ aufweisen.*

Durch die Regelung der Vereinfachung soll erreicht werden, dass die Verhaltensabweichung von Teilmodellen in einem Toleranzrahmen verbleibt und diese somit valide sind. Die Verhaltensabweichung soll durch den Vergleich des Verhaltens eines durch Vereinfachung gebildeten Teilmodells $b_{i,h}$ und der entsprechenden Partition $p_{i,h}$ des Ausgangsmodells gemessen werden.

Anforderung 15 (Kardinale Skalierung der Messung der Verhaltensabweichung). *Zur einfachen Vorgabe von Zielwerten und Regelung soll das Maß zur Erfassung der Verhaltensabweichung kardinal skalierte Ergebnisse liefern.*

2.2.5 Anforderungen an die Generierung von Zustandsabbildungen

Zur automatischen Erstellung von Zustandsabbildungen soll ein Algorithmus entwickelt werden, der aus den während der Vereinfachung durchgeführten Operationen eine Abbildung ableitet und damit einen konsistenten und ähnlichen Zustand für ein Teilmodell $b_{i,h}$ aus den komplexeren Teilmodellen $B_j(b_{i,h})$ berechnen kann und umgekehrt.

⁷ $b_{i,h}$ ist das Teilmodell i auf Hierarchieebene h und $p_{i,h}$ die Partition des Ausgangsmodells, die von $b_{i,h}$ abgebildet wird.

Anforderung 16 (Qualität der Zustandsabbildungsgenerierung). *Die erzeugten Abbildungen müssen während der Durchführung einer dvdS solche Zustände generieren, dass sich der Zustand der dvdS nach der Rekombination möglichst wenig von dem unterscheidet, den sie ohne Rekombination gezeigt hätte.*

2.2.6 Anforderungen aus der Integration des Verfahrens in d³FACT

Die in dieser Arbeit zu entwerfende geregelte Vereinfachung soll keine allgemeingültige Vereinfachung für alle Ziel- und Aufgabenstellungen der Materialflusssimulation sein. Die durch Vereinfachung erzeugten Teilmodelle sollen innerhalb des Simulators d³FACT im Rahmen einer dvdS verwendet werden können, um Problemstellungen der Materialflusssimulation lösen zu können.

Anforderung 17 (Integrationsfähigkeit in d³FACT). *Die erzeugte Hierarchie von Teilmodellen muss die Ausführung einer dvdS zur Bearbeitung von Aufgaben der Materialflusssimulation im Simulationswerkzeug d³FACT erlauben.*

3 Stand der Technik

Standing on the
Shoulders of Giants.

(Oasis)

In diesem Kapitel sollen bestehende Ansätze untersucht werden, die sich mit der Lösung der aus Abschnitt 2.1 bekannten Teilprobleme Modellbeschreibung, Partitionierung und Hierarchisierung, Modellvereinfachung und dessen Regelung, Zustandsgenerierung und dem Simulationswerkzeug d³FACT befassen. Diese Ansätze sollen insbesondere darauf überprüft werden, ob sie zur Erfüllung der an die Problemlösung gestellten Anforderungen des Abschnitts 2.2 gänzlich oder teilweise geeignet sind.

3.1 Modellbeschreibungsmethoden in der Materialflusssimulation

In diesem Abschnitt sollen nach einer allgemeinen Übersicht über Modellbeschreibungsmethoden drei spezielle Modellbeschreibungsmethoden detailliert dargestellt werden. Zum einen ist dies die Modellbeschreibungsmethode der dvdS [Mue05], weil diese Arbeit auf der dvdS aufbaut und diese erweitert. Aus dem gleichen Grund wird die Modellbeschreibungsmethode des Simulationswerkzeugs d³FACT vorgestellt. Abschließend werden kurz die Grundlagen der Discrete Event Simulation Specification (DEVS) vorgestellt, damit ein besseres Verständnis für einige Arbeiten zur Modellvereinfachung gegeben ist.

3.1.1 Modellbeschreibungsmethoden im Allgemeinen

Eine Sammlung und Klassifikation von Methoden zur Modellierung von Materialflüssen geben bspw. Mertins et al. [MSJ93] an. Sie unterscheiden zwischen Modellen und Modellie-

rungsmethoden, die Sprachkonstrukte und Vorgehensweisen zur Erstellung von Modellen enthalten. Modellierungsmethoden unterteilen sie weiter in Methoden zur Beschreibung von Aspekten von Produktionssystemen, Methoden zur Systemanalyse und zum Softwareentwurf und in anwendungsneutrale Beschreibungsmethoden, die neben dem Einsatz zur Modellierung von Produktionsprozessen weitere Anwendungsfelder haben.

Aufbauend auf diesen Methoden existiert eine Reihe von handelsüblichen Materialflusssimulatoren [MD03, KG99] (z. B. Simul8, Quest oder Plant Simulation). Diese Programme geben dem Modellierer auf eine spezielle Anwendung oder Domäne zugeschnittene Werkzeuge und Komponenten an die Hand. Modelle können durch Kombination, Parametrisierung und Verbindung von Komponenten aus einer Bibliothek auf einer meist graphischen Oberfläche erstellt werden. Diese Modelle bilden Module und können als neue Komponenten für die Bibliothek dienen. Werden diese in ein Modell eingesetzt, so entsteht eine Hierarchie aus Modulen/ Komponenten, die wiederum Module/ Komponenten enthalten. Durch derartige Hierarchien können auch große, komplexe Modelle erstellt und gewartet werden [SW00].

3.1.2 Die Modellbeschreibungsmethode der dvdS

Die Modellbeschreibungsmethode von Mueck [Mue05] orientiert sich an den handelsüblichen Simulatoren (s.o.). Zusätzlich sollte eine Visualisierung des Modells in einer virtuellen Realität (VR) erfolgen können, eine Eigenschaft, die seinerzeit von wenigen der handelsüblichen Simulatoren angeboten wurde. In der dvdS werden Zustände durch die Markenbelegung von Positionen (Komponenten) definiert. Neben der Belegung mit Marken, die nicht unterscheidbar sind, enthält das Modell eine Menge von Ereignissen E zum Erzeugen, Vernichten oder Verschieben der Marken, durch die ein Fortschreiten des Simulationszustands beschrieben wird. Zur Ereignisbehandlung enthält ein Modell eine Übergangsfunktion d , die, abhängig von dem Ereignis und dem Zeitpunkt, die Anzahl an Marken an der zugeordneten Position bestimmt und eine Menge neuer Ereignisse einplant. Um eine Position o in einer VR zu visualisieren, ist eine Definition ihres Ortes p nötig, an dem die Form f angezeigt werden soll.

Folgend soll die Modellbeschreibung von einem dvdS-Modell eingehend dargestellt werden. In einer dvdS existieren Positionen auf mehreren Detailebenen. Eine Position o ist bspw. auf drei Detailebenen vorhanden, aber nur eine dieser drei „Parallelpositionen“ ist

im Modell eingebunden und wird simuliert. Wenn der Komplexitätsgrad der Position o^i verändert werden soll, dann muss o^i aus dem Modell entfernt und $o^{i\pm 1}$ eingesetzt werden. Der Zustand von o^i muss dazu mit der Übergangsfunktion ψ auf $o^{i\pm 1}$ übertragen werden. In Abschnitt 3.5 wird detailliert auf das Erzeugen neuer Zustände durch Übergangsfunktionen eingegangen.

Ein detaillierungsvariantes diskretes Simulationsmodell dvdS ist ein Tupel

$$dvdS = (O, F, P, d, \phi, T, E, \vec{h}, \vec{S}_0, d_0) \quad (3.1)$$

mit

$O = (o_1, \dots, o_n)$	Menge der Positionen
$F = (f_1, \dots, f_n) \subset \mathbb{R}^3$	Menge der Formen für die Position
$P = (p_1, \dots, p_n) \in (\mathbb{R}^3)^n$	Orte, an denen die Positionen visualisiert werden sollen
T	Menge der möglichen Zeitpunkte
E	Menge der verschiedenen Ereignisse
$\delta : (E \times O \times T) \rightarrow (Z \times (E \times O \times T)^* \times (\hat{E} \times \hat{O} \times \hat{T})^*)$	Ereignisbehandlung
$(\hat{E} \times \hat{O} \times \hat{T})$	Menge der Ereignisbehandlungen des nächstgrößeren Modells
$\phi : (\hat{E} \times \hat{O} \times \hat{T}) \rightarrow (E \times O \times T)$	Behandlung von Ereignissen, die von einem Modell anderer Detaillierung ausgelöst werden
$\vec{h} \in ((M', F', P', O', \delta', \phi', T', E', \hat{h}', \hat{S}'_0, \delta'_0) \mid \{\})^n$	Vektor mit detaillierteren Modellen
$\vec{S}_0 \in \mathbb{N}_0^n$	Startbelegung der Positionen
$\delta_0 \in (E \times O \times T)$	Startereignis

Existiert für eine Position o_i ein detailliertes Modell, dann wird es in $\vec{h} = (h_1 \dots h_n)$ mit h_i beschrieben. Existiert kein detailliertes Modell für o_i , dann ist $h_i = \emptyset$.

3.1.3 Die Modellbeschreibungsmethode von d³FACT

Der Simulator d³FACT, der von Laroque [Lar07] entwickelt wurde, greift viele Eigenschaften der Modellierungsmethode von Mueck und von Petri-Netzen auf. Modelle in d³FACT sind ebenfalls für eine Visualisierung in einer virtuellen Umgebung vorgesehen. Ein d³FACT Modell ist:

$$DFM = (V, \Psi, s_0(t_0)) \quad (3.2)$$

mit

$v = (I^v, o, f, Ch^v, E^v) \in V$	Menge der Komponenten
$I^v \subset I$	Werte der Zustandsvariablen von v
$o \in \mathbb{R}^3$	Position von v
$f \subset \mathbb{R}^3$	Form von v
$Ch^v \subset Ch$	Menge der Channels von v
$E^v \subset E$	Menge der Ereignisroutinen von v
$\Psi = ch \times ch \rightarrow \{0, 1\}$	Adjazenzmatrix der Channel
$s_0(t_0)$	Ausgangszustand
$s_j(t_i) = (Q_j(t_i), I_j(t_i), M_j(t_i))$	Der j-te Zustand zum Zeitpunkt t_i
$I_j(t_i)$	Die j-ten Werte aller Variablen zum Zeitpunkt t_i
$M_j(t_i)$	Die j-te Markenbelegung zum Zeitpunkt t_i
(e, t)	Für den Zeitpunkt t eingeplante Ereignisroutine e
$Q_j(t_i) = \{(e, t) \mid t \geq t_i\}$	Die j-te Menge aller zum Zeitpunkt t_i eingeplanten Ereignisroutinen
$\delta : s_j(t_i) \xrightarrow{(e, t_i)_j} s_{j+1}(t_i)$ $Q_{j+1}(t_i) = Q_j(t_i) \setminus (e, t_i)_j$ $t_i = \min(t \mid (e, t) \in Q_{j+1}(t_i))$	Ereignisbehandlung für alle $j= 1 \dots k$ $(e, t)_j$ zum Zeitpunkt t_i

Die logistische Verknüpfung von Komponenten erfolgt über die Verknüpfung von Channels, die jede Komponente hat. Über die Verknüpfungen werden Marken durch Ereignisse verschickt und empfangen. Da zu einem Zeitpunkt mehrere Ereignisse ausgeführt werden

können, können auch mehrere ($j = 1..k$) Zustände zu einem Zeitpunkt existieren. Ein Zustand in d³FACT ist festgelegt durch die Werte der Variablen, der Markenbelegung und der Menge aller eingeplanten Ereignisroutinen. In der Ereignisbehandlung werden alle zu einem Zeitpunkt eingeplanten Ereignisroutinen sequentiell ausgeführt und aus der Menge der eingeplanten Ereignisroutinen gelöscht. Die Ausführung einer Ereignisroutine bildet einen neuen Zustand, indem die Zustandsvariablen und die Markenbelegung verändert werden und der Menge der eingeplanten Ereignisroutinen neue Elemente hinzugefügt werden. Sind alle Ereignisroutinen abgearbeitet, die für den aktuellen Zeitpunkt eingeplant sind, so springt die aktuelle Zeit auf die Zeit der am frühesten eingeplanten Ereignisroutine in Q .

3.1.4 Die Modellbeschreibungsmethode der Discrete Event System Specification

Ein DEVS-Modell ist ein Tupel

$$M = (X, Y, S, \delta, \lambda, \tau)$$

mit der Menge der externen Ereignisse X , der Menge der sequentiellen Zustände S , der Menge der Ausgabewerte Y , der Transitionsfunktion δ , der Ausgabefunktion λ und der Zeitfortschrittsfunktion τ [Foo01].

$$\tau : S \rightarrow R^+$$

$$Q = (s, e) \mid s \in S, 0 \leq e \leq \tau(s)$$

$$\delta : Q \times (X \cup \{\#\}) \rightarrow S$$

$$\lambda : Q \rightarrow Y$$

$\tau(s)$ ist die Zeit, für die das System im Zustand s sein wird. Das Paar (s, e) gibt an, dass das System im Zustand s für die Zeit e ist. Das $\#$ in der Transitionsfunktion δ bezeichnet, dass kein externes Ereignis erfolgte. Die autonome Transitionsfunktion $\delta(s, e, \#) = \delta_{\#}(s, e)$ bringt das System in den Zustand s , bis dieser nach der Zeitdauer $\tau(s)$ durch eine weitere Transitionsfunktion verlassen und Zustand s_1 mit $(s_1, 0)$ und $\tau(s_1)$ eingestellt wird. Vor der Zustandsänderung wird die Ausgabefunktion $\lambda(s)$ ausgeführt. Wenn ein externes Ereignis x eintritt, dann stellt sich Zustand $(s', 0)$ ein, definiert durch $\delta(s, e, x)$. Die externen

Ereignisse - die Eingaben in das Modell - und die Ausgaben haben einen Zeitindex und bilden somit Trajektorien.

3.2 Methoden der Partitionierung und Hierarchisierung

3.2.1 Partitionierungsalgorithmen

Das klassische Partitionierungsproblem besteht dabei darin, die Knoten eines Graphen in mehrere Teilmengen zu teilen. Dabei wird die Anzahl der geschnittenen Kanten unter der Restriktion minimiert, so dass jede Teilmenge gleich viele Knoten (bzw. gleich viel Knotengewicht) enthält. Graphpartitionierungsprobleme sind heute in vielen Anwendungsbereichen vorzufinden, wie bspw. in der Parallelisierung von Computerprogrammen oder der Finite Elemente Methode.

Das Partitionierungsproblem definiert sich wie folgt:

Gegeben ist der Graph $G = (V, K)$ (mit V als Menge von gewichteten Knoten und K als Menge der gewichteten Kanten), dann heißt

$$p = \{V_1, V_2, \dots, V_n\} \text{ mit } \bigcup_{i=1}^n V_i = V \text{ sowie } V_i \cap V_j \neq \emptyset \forall i \neq j$$

n -Partition von G ¹. Im Spezialfall $n = 2$ wird p eine Bisektion von G genannt. Die Funktion

$$bal(p) := \max \left\{ \omega(V_i) - \frac{\omega(V)}{n}; 0 < i \leq n \right\}$$

wird als Balance der Partition p bezeichnet. Eine Bisektion ist balanciert, wenn gilt $bal(p) < \max\{\omega(v); v \in V\}$ [Pre01]. Die Menge $\mathcal{S} = \{(u, v) \in K | P(u) \neq P(v)\}$ heißt geschnittene Kantenmenge von p . Die Schnittgröße von p ist

$$cut(p) := \sum_{(v,w) \in \mathcal{S}} \omega((v, w))$$

¹In der Literatur wird üblicherweise von p -Partition gesprochen. Da in dieser Arbeit der Bezeichner p anders belegt ist, wird stattdessen die Notation n -Partition verwendet.

Die Berechnung einer optimalen Lösung für das Graphpartitionierungsproblem ist NP-Vollständig (siehe bspw. [GJ79]), deshalb werden Heuristiken zur Lösung des Problems eingesetzt. Sie lassen sich nach folgenden Kriterien unterteilen:

- Deterministische Methoden ggü. Randomisierte Methoden
- Sequentielle Methoden ggü. Parallele Methoden
- Single-Level-Methoden ggü. Methoden mit Multilevel-Ansatz
- Globale Methoden ggü. Lokale Methoden

Deterministische Methoden liefern für einen gegebenen Graphen bei jedem Durchlauf dasselbe Ergebnis, während andere Methoden zufallsabhängige Elemente enthalten können und so bei einem gegebenen Graphen in jedem Versuch verschiedene Ergebnisse liefern. Partitionierungsmethoden können sequentiell implementiert werden oder durch Parallelisierung der Prozesse, bei entsprechender Hardware, schneller Ergebnisse liefern. Im Rahmen dieser Arbeit werden nur sequentielle Methoden betrachtet, da die Laufzeit des Algorithmus, wie bereits beschrieben, nur eine untergeordnete Rolle spielt und sequentielle Methoden grundsätzlich einfacher zu analysieren und zu implementieren sind.

Single-Level-Ansätze versuchen den Graphen in seinem ursprünglichen Zustand zu partitionieren, während Multilevel-Methoden den Graphen zunächst stark vereinfachen, ihn in diesem vereinfachten Zustand partitionieren und schließlich wieder rekonstruieren.

Globale Methoden versuchen den Graphen als Ganzes zu partitionieren. Lokale Methoden arbeiten eher auf der Ebene von einzelnen Knoten und versuchen sich an eine optimale Partition heranzutasten, indem Knoten lokal zwischen den Partitionen ausgetauscht werden. Globale und Lokale Methoden ergänzen sich oft, dabei liefert ein Globaler Algorithmus eine grobe Ausgangslösung, die dann durch einen lokalen Verbesserungsalgorithmus optimiert wird.

Im Folgenden soll eine kurze Übersicht über für diese Arbeit wichtige Methoden gegeben werden, als umfassende Übersichten sind [Fjä98, Els97, AK95, Pre01] zu empfehlen.

Lokale Verbesserungsmethode: Kernighan-Lin

Diese lokalen Verbesserungsmethoden versuchen durch lokale Knotenverschiebungen eine bereits vorgegebene Ausgangslösung zu optimieren. Die meisten Algorithmen arbeiten dabei auf Grundlage einer Bisektion. Ein rekursiver Aufruf lokaler Algorithmen dient dann

dazu, eine n -Partition zu erreichen. Der Nachteil solch eines indirekten Vorgehens sind i.d.R. schlechtere Ergebnisse als bei einer sofortigen n -Partitionierung [DMP94, Els97].

Kernighan und Lin [KL70] haben einen der frühesten Ansätze zur Partitionierung von Graphen vorgestellt. Die Kernighan-Lin-Methode (KL) geht dabei von einer bereits vorhandenen Bisektion aus und versucht diese durch eine Reihe von Vertauschungen der Knoten zu verbessern. Das angestrebte Ziel ist dabei die Minimierung der Schnittgröße der Bisektion. Die KL-Methode berücksichtigt explizit Graphen mit gewichteten Knoten. Zum Zweck der besseren Lesbarkeit wurde das Knotengewicht hier nicht weiter berücksichtigt bzw. gleich 1 gesetzt. Gegeben sei der Graph $G = (V, K)$ und eine Bisektion $p = \{P_1, P_2\}$ von G und eine gewichtete Adjazenzmatrix $A = (a_{ij})$. Für jeden Knoten $v \in V$ werden auf der gegebenen Bisektion die internen und externen Kosten definiert.

$$\begin{aligned} cint(v) &= \sum_{(v,u) \in K; P(u)=P(v)} \omega(v, u) \\ cext(v) &= \sum_{(v,u) \in K; P(u) \neq P(v)} \omega(v, u) \end{aligned}$$

Der Nutzenzuwachs, der entsteht, wenn man einen Knoten v in eine andere Partition verschiebt, ist dann definiert als $gain(v) = cext(v) - cint(v)$ und der Nutzenzuwachs durch das Vertauschen von zwei Knoten $gain(v_i, v_j) = gain(v_i) + gain(v_j) - 2a_{ij}$.

Gegeben sei eine beliebige Bisektion $p = \{P_1, P_2\}$, so existieren zwei Mengen $X \subset P_1$ und $Y \subset P_2$, durch deren Vertauschung man die minimale Bisektion $p^* = \{P_1^*, P_2^*\}$ erhält. Der KL-Algorithmus versucht nun die Mengen X und Y sequentiell zu approximieren. Eine Iteration des ursprünglichen KL-Algorithmus terminiert in $O(|V|^3)$. Eine Verbesserung dieser Laufzeit auf $O(|K| \max\{\log|V|, deg_{max}(G)\})$ wird von Dutt [Dut93] vorgestellt.

Globale Methode: Greedy

Obwohl es auch lokale Greedy-Algorithmen gibt, wird diese Methode hier als globale Methode angewendet, weil Greedy-Algorithmen sinnvoll sein können, wenn eine Partition mit nur wenig Zeitaufwand bestimmt werden muss, die dann als Grundlage für lokale Verbesserungsalgorithmen dienen kann. Ein Greedy-Algorithmus trifft immer eine Entscheidung, die in der aktuellen Situation den höchsten Nutzenzuwachs hat, so dass mit hoher Wahrscheinlichkeit nur ein lokales Optimum, aber nicht ein globales erreicht wird. Wegen dieses

einfachen Vorgehens sind Greedy-Algorithmen i.d.R. sehr klein und schnell in der Ausführung. Im Vergleich zu anderen Algorithmen ist die Ergebnisqualität allerdings eher minderwertig, da sie lokale Optima nicht verlassen können. Einfache Algorithmen für eine p-Partitionierung sind bspw. in [Els97, Pot97] beschrieben.

Battiti und Bertossi [BB98] stellen den Greedy-Algorithmus Min-Max-Greedy vor, der bei der Auswahl der Knoten etwas bedächtiger vorgeht. Zuerst sucht er alle freien Knoten mit minimaler Anzahl an partitionsexternen Nachbarknoten und speichert sie in einer Liste. Aus dieser Liste wird der Knoten ausgewählt, der die maximale Anzahl an partitionsinternen Nachbarknoten aufweist. Zudem wird eine Verbesserung dieses Algorithmus vorgestellt: Differential-Greedy modifiziert die Vorgehensweise für die Auswahl des hinzuzufügenden Knotens. Hierbei wird die zweistufige Auswahl des Min-Max-Algorithmus durch eine der KL-Methode ähnelnde Berechnung der gain-Werte substituiert. Der Knoten mit der minimalen Differenz zwischen partitionsexternen und partitionsinternen Nachbarknoten wird als nächstes in die Partition einbezogen. Gilt $|V| \leq |K|$, dann terminiert dieser Greedy-Algorithmus in $O(|K|)$.

Multilevel Ansatz

Der Begriff der Multilevel Ansätze steht eher für eine Rahmenumgebung zum Einsatz vorhandener Partitionierungsalgorithmen als für einen eigenständigen Algorithmus. Der Gedanke hinter ML-Algorithmen ist es, den Graphen unter der Beibehaltung seiner Struktureigenschaften über mehrere Ebenen so weit zu vereinfachen, bis dieser mit bekannten Heuristiken oder sogar mit mathematisch exakten Optimierungsverfahren in hinreichend kurzer Zeit partitioniert werden kann. Der Vereinfachungsschritt wird im Folgenden Vergrößerung genannt. Nachdem der vereinfachte Graph partitioniert wurde, wird die dabei entstandene Partition wieder auf den ursprünglichen Graphen zurück projiziert. Diese Verfeinerung läuft analog zur Vergrößerung über mehrere Ebenen, wobei auf jeder Ebene zusätzlich ein lokaler Verbesserungsalgorithmus eingesetzt werden kann, um die Partition weiter zu optimieren. ML-Ansätze zielen darauf ab, dass die für die Vergrößerung, Partition und Verfeinerung des Graphen verbrauchte Zeit kleiner ist als die Zeit, die der entsprechende herkömmliche Algorithmus – bei vergleichbarem Ergebnis – für die Partitionierung des großen, ursprünglichen Graphen benötigen würde. Dieser Gedanke ist daher besonders für Algorithmen mit nichtlinearer Laufzeit interessant.

Das Vorgehen von ML-Ansätzen wird von Karypis und Kumar [KK95] beschrieben. Die erste Phase sucht schrittweise nach Vergrößerungen des Graphen. Um eine solche zu erhalten, wird für den Graphen ein (möglichst) maximales Matching berechnet. Eine Kantenmenge $K_M \subseteq K$ ist ein Matching, wenn keine zwei Kanten aus K_M einen gemeinsamen Endknoten besitzen. Ein maximales Matching ist ein Matching mit maximaler Kantenzahl. Ein Knoten v heißt bezüglich des Matchings K_M gesättigt, wenn dieser Knoten Endknoten einer Kante des Matchings ist. Je zwei, durch eine Kante des Matchings gesättigte Knoten werden dann zu einem Metaknoten verschmolzen. Der Metaknoten hat das Knotengewicht, das der Summe der Gewichte der zusammengefassten Knoten entspricht. Kanten, die zuvor von den einzelnen Knoten ausgingen, gehen jetzt von dem Metaknoten aus. Dabei entstehende parallele Kanten werden analog zu den Knoten zusammengefasst und Schlingen werden entfernt. Die Bestimmung eines optimalen Matchings ist nicht in linearer Laufzeit lösbar [Gab90], allerdings existieren Approximationen für lineare Laufzeit [Pre01, PS04]. Karypis und Kumar stellen zwei Matching-Methoden vor, das Heavy Edge Matching und das Heavy Clique Matching, wobei letzteres bei ihren Untersuchungen bessere Ergebnisse lieferte. Alternativ zum Matching können auch andere Methoden eingesetzt werden, die Knoten nach bestimmten Merkmalen gruppieren. Solche Methoden können beispielsweise agglomerative Clustering-Verfahren oder unabhängige Knotenmenge sein [AK95, BS94].

In der zweiten Phase wird auf den vereinfachten Graphen ein herkömmlicher Partitionierungsalgorithmus angesetzt. Ein alternativer Ansatz, der diesen Schritt unnötig macht, ist es, den Graphen so weit zu vereinfachen, bis dieser nur noch n Knoten enthält, die dann als Ausgangsknoten für eine n -Partition dienen [Gup97]. Die dritte Phase führt den Graphen durch eine schrittweise Verfeinerung zurück zum Ausgangsgraphen. Wird dabei ein Metaknoten \hat{v} wieder in zwei Teile geteilt, so behalten beide Unterknoten v_1 und v_2 die Partitionszugehörigkeit des Metaknotens. Während der Vergrößerung kann es wünschenswert sein, die Partitionen mit lokalen Verbesserungsverfahren zu optimieren, dazu kann bspw. ein Greedy Refinement genannter vereinfachter KL-Algorithmus eingesetzt werden [KK98b, KK98a].

Saab [Saa97] untersucht insbesondere die Effektivität des Vergrößerungsschritts in Hinblick auf die Ergebnisqualität. Verglichen mit den Ergebnissen desselben Optimierungsalgorithmus, der auf den ursprünglichen Graphen angewendet wurde, verbesserte der vorgeschaltete Vergrößerungsschritt das Ergebnis abhängig vom durchschnittlichen Knotenrang. Insbesondere bei Graphen mit niedrigen Knotenrängen verbesserte sich das Ergebnis

so um 37 bis 74 Prozent.

Karypis und Kumar stellen ihre Methoden ausführlich dar und hinterlegen sie mit Analysen und Bewertungen, die einen erfolgreichen Einsatz einer Multilevel-Methode mit Matching und einem lokalen Verbesserungsverfahren im Verfeinerungsschritt in dieser Arbeit versprechen.

3.2.2 Methoden der Hierarchisierung

Weil Simulationsmodelle im Durchschnitt immer komplexer werden, besteht die Notwendigkeit, Modellteile in weiten Teilen wiederverwendbar zu machen. Dies geschieht, um den zeitaufwendigen Modellierungsprozess zu verkürzen. Werden Modelle so erstellt und abgelegt, dass sie in anderen Modellen wiederverwendet werden können, so heißen sie Module. Die Verwendung von Modulen hat zudem den Vorteil, dass das Modellverständnis durch eine Strukturierung verbessert wird [PBC98].

Eine Möglichkeit, eine Modellhierarchie auch mit Modellen unterschiedlicher Komplexität zu verwalten und die Modellkomposition zu automatisieren, bietet das System Entity Structure/Model Base Framework (im Folgenden mit SES/MB bezeichnet) [KLCZ92, ZPK00]. Das SES/MB versucht zur Komposition von Modellen einen Baum über die möglichen Teilmodelle aufzubauen. Um dies zu gewährleisten, enthält das System eine Model Base, in der die Menge aller verfügbaren Modelle/Teilmodelle abgelegt ist, und eine Entity Structure Base, die die Komposition von Modellen aus Teilmodellen der Model Base beschreibt. Dieser Baum - die Entity Structure Base - enthält Knoten, die auf Teilmodelle aus der Model Base verweisen. Zwischen diesen Modellen stehen Beschreibungen, wie die komplexeren Modelle zum jeweils weniger komplexen verbunden werden müssen. An einem Knoten können zwei Möglichkeiten hinterlegt werden, ein dem Knoten entsprechendes Modell zu erstellen: eine Referenz auf Modelle aus der Model Base und eine Komposition aus Knoten, die wiederum für Modelle stehen, die Kinder von dem betrachteten Knoten sind. Soll nun ein neues Modell erstellt werden, kann ein Baum entlang der Entity Structure Base aufgestellt werden, der alle Möglichkeiten beschreibt, das Modell zusammenzustellen (siehe Abbildung 3.1). Potentiell bisher nicht vorhandene, aber benötigte Teilmodelle erscheinen als Blätter an dem Baum und müssen vor Generierung des Modells der Model Base hinzugefügt werden. Durch den Einsatz des Baumes entsteht so eine Modellfamilie, die Modelle in unterschiedlichen Detaillierungsgraden enthält. Soll

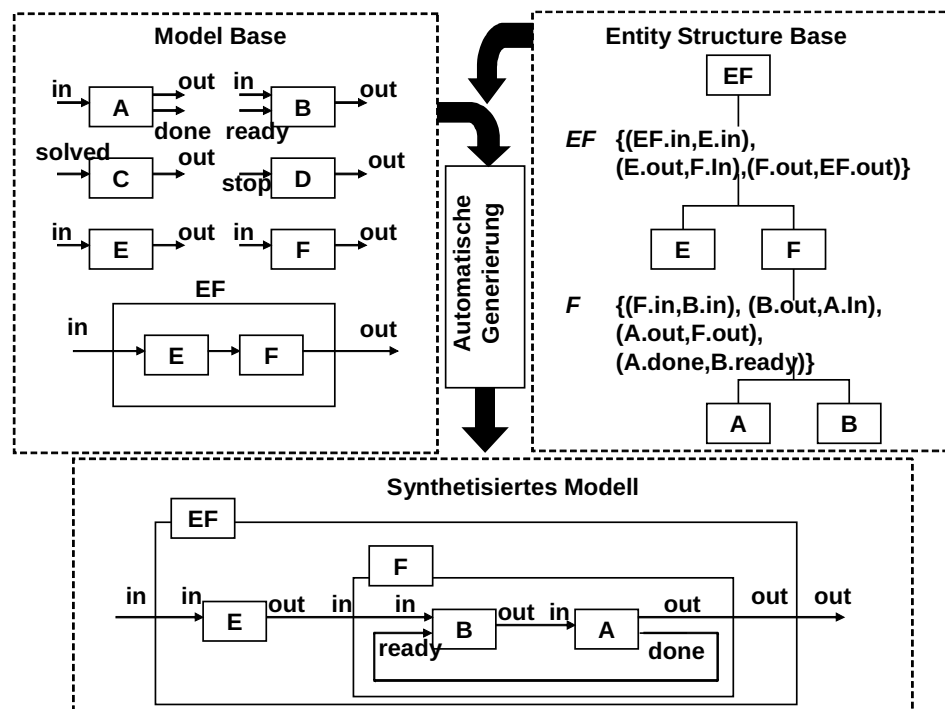


Abbildung 3.1: Ein Beispiel für das System Entity Structure/Model Base Framework

nun eine Simulation ausgeführt werden, muss der Modellierer je nach Einsatzzweck festlegen, welche Teilmodelle er verwenden möchte. Ein Simulationsmodell kann basierend auf dem Baum, der Model Base und den Festlegungen des Modellierers automatisch generiert und ausgeführt werden.

Der Ansatz erlaubt, Modelle von einem System in unterschiedlichen Komplexitätsgraden zu verwalten und diese für eine Wiederverwendung zu selektieren und zu kombinieren. Des Weiteren können durch diesen Ansatz bei der Erstellung eines neuen Modells Lücken in der Modellfamilie schnell gefunden werden. Der Ansatz zielt damit auch auf das Problem, dass für eine Wiederverwendung von Modellen diese in einem passenden Komplexitätsgrad vorliegen müssen.

3.3 Methoden der Vereinfachung von ereignisdiskreten Simulationsmodellen

Die Literatur im Bereich der Vereinfachung von diskreten Simulationsmodellen lässt sich in zwei Bereiche teilen: die Festlegung von grundsätzlichen Zielen und Methoden und

Verfahren, die durch die Implementierung der Methoden diese Ziele erreichen wollen. In den folgenden Abschnitten werden alle relevanten Arbeiten in beiden Bereichen dargestellt und auf ihre Eignung zur geregelten Vereinfachung hin untersucht.

3.3.1 Grundlegende Ziele und Methoden

Zeigler und andere [ZPK00, Fra95, Sev91] definieren Modellvereinfachung als Reduzierung der Komplexität eines Modells, wobei der u.U. erzeugte Verhaltensunterschied des Modells innerhalb des experimentellen Rahmens tolerierbar bleibt.

Frantz² [Fra95] stellt eine umfassende Taxonomie von Vereinfachungsmethoden vor (siehe Abbildung 3.2). Darin unterteilt er alle vorhandenen Verfahren in drei Gruppen: die Modifikation und Beschränkung der Einflussgrößen, die Modifikation des Modellverhaltens und die Modifikation des Modellierungsformalismus.

Durch Beschränkung der Einflussgrößen sollen Berechnungen oder Abläufe vereinfacht werden. Beispielhaft wären die Vernachlässigung des Stauverhaltens auf Fördersystemen oder die Rüstzeiten von Maschinen zu nennen. Die Technik der Verwendung einer Modellhierarchie (in Abbildung 3.2 dunkel hinterlegt) kann in dieser angewendet werden, indem Komponentenklassen unterschiedlicher Komplexität erzeugt werden³. Dies können bspw. Anlagen mit und ohne Berücksichtigung von Rüstvorgängen oder Förderbänder mit und ohne Berücksichtigung von Stauverhalten sein.

Zur Modifikation des Modellverhaltens können gewisse Aspekte eines Modells zusammengefasst werden. Diese Aspekte sind Zustände, Zeitfunktionen, Entitäten und Funktionen. Es können Zustände zusammengefasst werden, um so die Zustandsmenge zu verringern, oder es kann die Zeitfunktion vereinfacht werden, indem die Granularität des Zeitfortschritts erhöht wird. Diese beiden Methoden sind nicht in dieser Arbeit anwendbar, weil Zustände kein expliziter Bestandteil des Modellierungsformalismus sind⁴ und die Zeit-

²Lee und Fishwick [LF96] stellen eine sehr ähnliche, wenn auch weniger detaillierte Taxonomie der Vereinfachung auf. Die Taxonomie von Lee und Fishwick ist eine Teilmenge von der von Frantz, deshalb soll hier auf eine Darstellung der Taxonomie von Lee und Fishwick verzichtet werden.

³Eine Hierarchie von Komponenten unterschiedlicher Komplexität muss erstellt werden, weil durch das Setzen der Rüstzeit auf 0 zwar erreicht werden kann, dass sich das Rüsten nicht mehr im Verhalten der Komponente widerspiegelt, aber die Komplexität der Komponente nicht geändert wird, da dieselbe Zahl und Art von Ereignissen ausgeführt wird.

⁴Diese Vereinfachungstechnik bietet sich für Modelle der DEVS an.

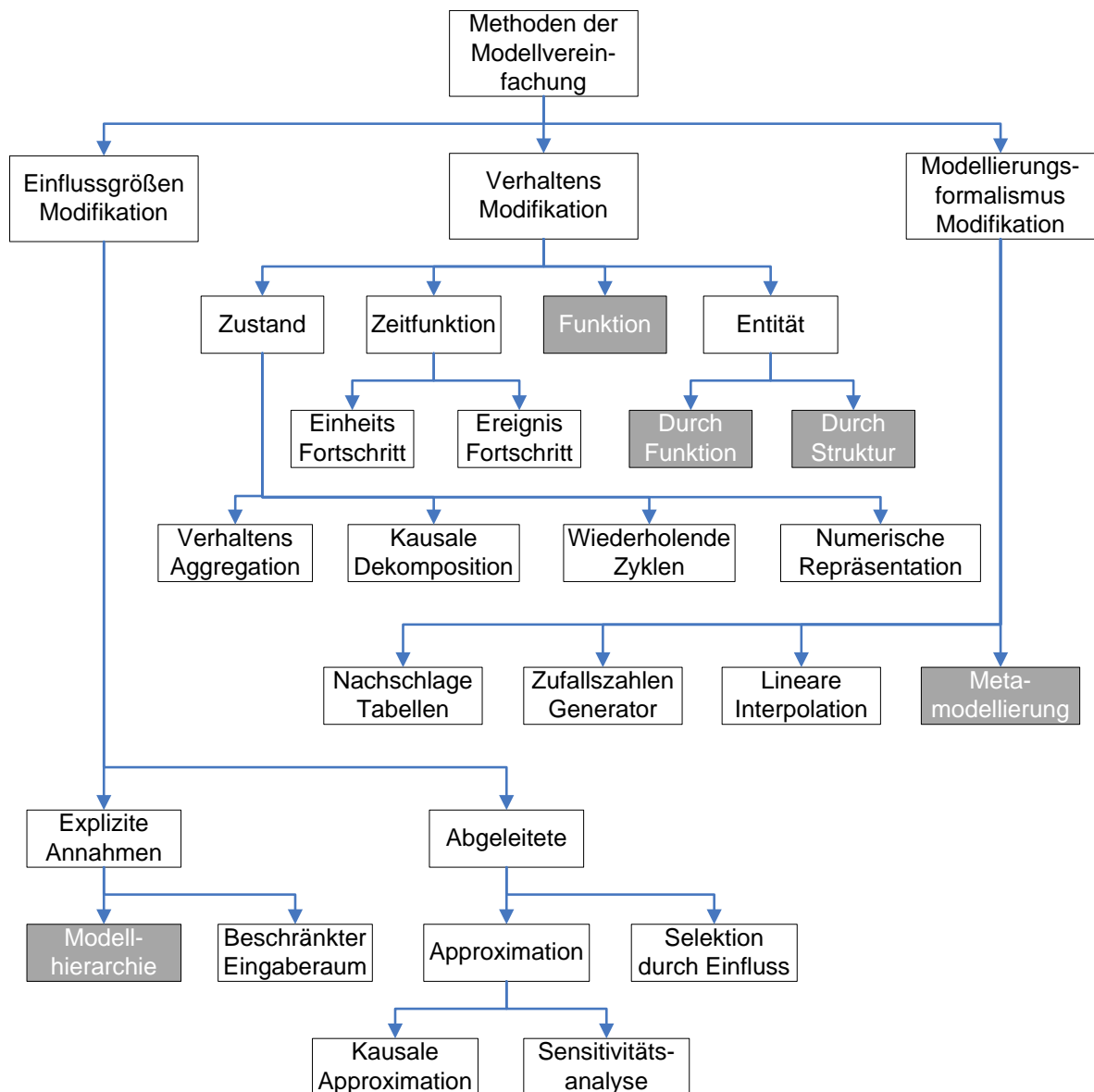


Abbildung 3.2: Taxonomie von Vereinfachungsmethoden [Fra95]

funktion nicht verändert werden kann, weil Teilmodelle unterschiedlicher Komplexität in einer Komposition gleichzeitig verwendet werden.

Die weiteren in dieser Arbeit benutzbaren Techniken sind die Zusammenfassung von Entitäten und Funktionen (in Abbildung 3.2 dunkel hinterlegt). Die Vereinfachung durch Aggregation ist eine übliche Methode, die besonders geeignet ist, wenn eine Modellhierarchie gegeben ist oder erzeugt werden kann. Höherstufige Komponenten bilden dann niedrigerstufige Komponenten auf einem höheren Abstraktionsgrad ab. Diese Aggregation kann zum einem nach dem Kriterium der Funktion durchgeführt werden, indem Komponenten mit gleicher oder ähnlicher Funktion zusammengefasst werden. Die Zusammenfassung hat dann eine Funktion, die der Summe aller Substituierten entspricht. Die Aggregation nach der Struktur ist schwer von der nach Funktion zu differenzieren. Hier kommt es nur darauf an, dass das Substitut eine nach außen hin gleiche Funktion aufweist, die internen Funktionen können andere als beim Ausgangsmodell sein. Bei der Vereinfachung nach Funktion bleiben die Entitäten bestehen, aber die Funktionen, die diese beinhalten, werden aggregiert.

Durch die Veränderung des Modellierungsformalismus wird die Funktion ausgetauscht, mit der das Modell Eingaben in Ausgaben abbildet. Die mächtigste dieser Methoden, die Metamodellierung, kann in dieser Arbeit zur Vereinfachung benutzt werden. Hier wird eine Black-Box erstellt, die das gleiche Ein- und Ausgangsverhalten aufweist wie das Ausgangsmodell⁵.

Zeigler [ZPK00] stellt grundsätzliche Methoden vor, mit denen die Komplexität reduziert werden kann, wobei die Methoden Aggregation, Linearisierung und der Wechsel des Modellierungsformalismus auch in der Taxonomie von Frantz vorhanden sind.

- Aggregation
Zusammenfassen von Bausteinen zu einem einzigen, welcher das Verhalten der substituierten aufweist. Die Anzahl Komponenten im Modell wird reduziert.
- Weglassung
Das Weglassen von Bausteinen, Variablen und Verknüpfungen führt ebenfalls zu weniger Komponenten im Modell.
- Linearisierung
Repräsentieren des Verhaltens um einen Arbeitspunkt als lineares System.

⁵Eine ausführliche Beschreibung von Metamodellen folgt in Abschnitt 3.3.4.

- Austauschen von deterministischen und stochastischen Variablen

Die aufwendige Berechnung einer Zufallszahl kann entfallen, wenn an ihrer statt bspw. der Mittelwert als Standard verwendet wird. Ist die Berechnung bzw. das Heraussuchen eines deterministischen Wertes aufwendig, so kann eine Zufallszahl den Berechnungsaufwand senken.

- Wechsel des Modellierungsformalismus

Verwendung eines anderen Formalismus, z.B. Differentialgleichungen oder Polynomielle Regression anstelle des ereignisdiskreten Modells.

Zeigler gibt keine nähere Erläuterung der Vereinfachungsmethode Linearisierung, deshalb wird sie als spezielle Form des Wechsels des Modellierungsformalismus angesehen (vgl. Frantz [Fra95]).

Zusammenfassend kann festgestellt werden, dass eine Reihe von Methoden zur Vereinfachung bestehen, die in dieser Arbeit einsetzbar sind. Wie sich in den nächsten Abschnitten zeigen wird, wurden diese Ideen auch von vielen anderen Autoren in der einen oder anderen Weise benutzt. Die bedeutendsten Methoden sind dabei die Aggregation von Komponenten und der Wechsel des Modellierungsformalismus.

3.3.2 Benutzung von Homomorphismen zur Vereinfachung

Ein Homomorphismus ist eine Abbildung h von einem Ausgangsmodell M zu einem vereinfachten Modell M' , so dass gilt [Foo74, Zei76]⁶:

$$\begin{aligned} h : S_M &\rightarrow S_{M'} \\ h(\delta_M(s, e, x)) &= \delta_{M'}(h(s), e, x) \\ \lambda_{M'}(h(s), e) &= \lambda_M(s, e) \\ h(\delta_M(s)) &= \delta_{M'}(h(s)) \\ t_{aM'}(h(s)) &= t_{aM}(s) \end{aligned}$$

Existiert eine solche Abbildung, dann weisen beide Modelle das gleiche Verhalten auf, die Vereinfachung M' ist demnach valide. Weil bei der Vereinfachung oftmals eine Abweichung

⁶Zeigler und Foo definieren Homomorphismen in dem Modellierungsformalismus DEVS, nähere Erläuterungen hierzu in [ZPK00] und in Abschnitt 3.1.4.

im Verhalten des Modells in Kauf genommen wird, wurden auch abgeschwächte Definitionen des Homomorphismus entwickelt [Sev90]. Mittels eines Homomorphismus kann also eine valide Vereinfachung von einem Modell berechnet werden.

Sevinc [Sev90, Sev91] benutzt einen abgeschwächten Homomorphismus, um ein Verfahren zur Vereinfachung zu erstellen. Dieses vereinfacht nicht atomare Teilmodelle zu atomaren Modellkomponenten (Zusammenfassung). Dazu wird für jedes Teilmodell ein Homomorphismus definiert, mit dessen Hilfe während eines Simulationslaufs alle Zustände und Funktionen vereinfacht werden. Durch einen Codegenerator können diese vereinfachten Modelle dann als eigenständige Modelle gespeichert werden. Einen schlüssigen Nachweis der Qualität und Funktionalität dieses Verfahrens bleibt der Autor allerdings schuldig. Auch müssen die zur Vereinfachung genutzten Homomorphismen manuell erstellt werden.

Die Verwendung von Homomorphismen nach Foo und Zeigler soll in dieser Arbeit nicht erfolgen, weil sie zum einen auf das nicht Verwendung findende DEVS ausgerichtet sind und zum anderen keine schlüssig dokumentierte Anwendung mit guten Ergebnissen bekannt ist.

3.3.3 Gezielte Reduzierung der Zahl ausmodellierter Anlagen oder Arbeitsgänge

Die Arbeiten von Brooks und Tobias [BT00], Johnson et al. [JFM05], Rose [Ros99, Ros00, Ros07a, Ros07b], Hung und Leachman [HL99], Jain et al. [JLGL99] und Völker [Völ03] benutzen alle ein sehr ähnliches Verfahren, um dasselbe Ziel zu erreichen, nämlich die Zahl der Ereignisse zu reduzieren, die benötigt werden, um einen Auftrag fertig zu stellen - eine Marke, ein Los, etc. bis zu einer Senke zu befördern. Um dies zu erreichen, wird die Anzahl an komplex modellierten Bearbeitungsschritten reduziert, indem die zu substituierenden Bearbeitungsschritte durch geschätzte Verweilzeiten ersetzt werden. So werden in den meisten Arbeiten komplex modellierte Anlagen durch deren durchschnittliche Bearbeitungszeit plus der durchschnittlichen Wartezeit vor den Anlagen ersetzt. Zentrale Bedeutung für die Validität der Vereinfachung kommt der Identifizierung der Engpässe oder des Engpasses im System zu. Die Engpässe werden während einer Simulation des Ausgangsmodells anhand der Auslastung der Anlagen oder der Streuung der Wartezeiten vor den Anlagen identifiziert. Diese Engpass-Anlagen werden dann nicht vereinfacht und

bleiben im Ausgangszustand erhalten. Nachfolgend sollen die wichtigen Ergebnisse aller Arbeiten in diesem Bereich kurz dargestellt werden.

Brooks und Tobias [BT00] arbeiten auf einer sehr modellnahen Ebene, so werden keine Simulationen zur Bestimmung der Engpässe durchgeführt, sondern diese anhand der effektiven Kapazität der Anlagen bestimmt. Eine Anlage besteht hier aus einem Ein- und einem Ausgangspuffer und einer zwischengeschalteten Maschine mit einer Bearbeitungszeit. Die effektive Kapazität $C_{effektiv}$ ergibt sich als:

$$C_{effektiv} = \frac{LS}{t_U} \cdot \frac{\text{Ruestintervall}}{t_R + \text{Ruestintervall}} \cdot \frac{MTBF}{MTTR + MTBF} \cdot \text{Ausschussquote} \quad (3.3)$$

mit

C	Kapazität
LS	Losgröße
t_U	Bearbeitungszeit
t_R	Rüstzeit
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair

Linear verknüpfte Maschinen werden durch eine Maschine ersetzt, die die Bearbeitungszeit der Engpass-Maschine aufweist und summierte Puffergrößen. Alle Puffer - inkl. einer Einheit für jede substituierte Maschine - vor dem Engpass werden zum Eingangspuffer der neuen Maschine addiert, analog dazu alle nachfolgenden zum Ausgangspuffer. Dadurch kann der Umlaufbestand die gleichen Werte annehmen, aber die Durchlaufzeit wird sich verändern. Um dies zu beheben, müssen den Puffern noch die summierten Bearbeitungszeiten der substituierten Maschinen als Verweilzeit zugewiesen werden. Eine ähnliche Vereinfachung kann auch bei parallelen Strukturen durchgeführt werden, was allerdings nicht weiter ausgeführt wird. In zwei Beispielen wurden diese Techniken zur Vereinfachung benutzt und es konnte bei einem Modell ohne Zufallseinflüsse eine Verbesserung der Laufzeit um 13 % und in einem Modell mit Zufallseinflüssen eine Verbesserung um 16 % erreicht werden. Vom Autor konnten keine Auswirkungen der Vereinfachung auf die Ergebnisqualität der Simulation festgestellt werden, was sehr zweifelhaft ist⁷.

Rose [Ros99, Ros00] vereinfacht Simulationsmodelle der Halbleiterfertigung, indem er alle Anlagengruppen bis auf den Engpass durch pauschale Verweilzeiten ersetzt. Aufgrund

⁷Eine Vereinfachung ohne Erzeugung eines Verhaltensunterschieds ist bei der Verwendung eines sinnvollen Ausgangsmodells nicht möglich, weil ansonsten die Erstellung komplexer Modelle sinnlos wäre.

der, für die Halbleiterindustrie typischen, Bearbeitungszyklen muss jedes Los den Engpass mehrmals belegen. So existiert sowohl vor dem Eingangspuffer und hinter der Anlage als auch im Rückfluss jeweils nur ein Element mit zufallsverteilter Verzögerungszeit. Zur Validierung wurden die Verteilungen der Durchlaufzeiten aus der Simulation des Ausgangsmodells mit der der Vereinfachung verglichen. Variiert wurde dabei die Auslastung der Engpassanlage und es wurden zwei Prioritätsregeln und zwei Einststeuerstrategien benutzt. Bei Verwendung der FIFO Regel als Prioritätsregel konnten gute Ergebnisse erzielt werden. Wurde hingegen mit einer auf der Schlupfzeit basierenden Regel gearbeitet, so zeigten sich systematische Fehler. Ein systematischer Fehler zeigte sich auch bei der Verwendung einer Einststeuerstrategie, die die Menge der Lose im System begrenzen sollte. Rose begründet diese systematischen Fehler damit, dass sich Lose im vereinfachten Modell im Gegensatz zum Ausgangsmodell relativ häufig überholen. Eine Angabe, inwiefern die Laufzeit durch die Vereinfachung reduziert werden konnte, bleibt Rose schuldig. Rose integriert in [Ros07a, Ros07b] weitere Verbesserungen in seinem Modell, um eine kleinere Abweichung in den Durchlaufzeiten zu erhalten. Das verzögernde Element im Rückfluss wird zum einen so erweitert, dass es eine auslastungsabhängige Verzögerungszeit aufweist, zum anderen soll das Überholen der Lose dadurch verhindert werden, dass das verzögernde Element durch ein Warteschlangensystem bestehend aus Puffer und Bedienstation ersetzt wird. Die Bedienstation kann unterschiedliche Verteilungen zur Bestimmung der Verzögerungszeit erhalten. Durch diese Maßnahmen konnte die Ergebnisqualität deutlich verbessert werden, wenn auch ein erkennbarer Unterschied erhalten bleibt.

Bei Johnson et al. [JFM05] wird ebenfalls in einer anfänglichen Simulation die Auslastung der Anlagen bestimmt und dann eine Rangfolge der Auslastungen erstellt. In einem iterativen Verfahren können nun Anlagen mit einer immer niedrigeren Auslastung durch pauschale Verweilzeiten substituiert werden. Bei jeder Iterationsschleife wird durch die Bestimmung des Korrelationskoeffizienten der Durchlaufzeiten von Ausgangsmodell und Vereinfachung bestimmt, ob die Verhaltensabweichung der Vereinfachung in Ordnung ist. Ist der Korrelationskoeffizient größer als 0,6, kann eine weitere Anlagengruppe durch eine Verweilzeit substituiert werden. Ist der Korrelationskoeffizient kleiner, so wird die letzte Vereinfachung, die einen Wert von über 0,6 erreichte, als endgültige Vereinfachung gewählt. Über den Korrelationskoeffizienten der Durchlaufzeiten kann somit die Vereinfachung gesteuert werden. Allerdings muss für dieses Vorgehen nach jeder Iteration eine neue Simulation des Modells erfolgen, so dass die Laufzeit der Vereinfachung bei komplexen Modellen sehr schlecht ist. In anschließenden Versuchen stellen Johnson et al. dar, dass es

Tabelle 3.1: Fehleranalyse zwischen Ausgangsmodell und Vereinfachungen [HL99]

Verweilzeit	Reduziertes Modell I		Reduziertes Modell II	
	deter- ministisch	stoch- astisch	deter- ministisch	stoch- astisch
Mittlerer Fehler der Durchlaufzeiten [%]	0,45	0,73	5,78	5,24
Standardabweichung der Abweichungen der Durchlaufzeiten [%]	1,11	1,72	16,37	23,34
Mittlerer Fehler der Anlagenauslastung [%]	-0,37	-0,36	-0,69	-0,48

von entscheidender Bedeutung ist, den Engpass des Systems nicht zu vereinfachen. Wird in einem Modell mit 10 sequentiellen M/M/10⁸ Systemen nur die Engpass-Anlagengruppe vereinfacht, so ist das Ergebnis von schlechterer Qualität, als wenn alle Anlagengruppen bis auf den Engpass vereinfacht werden. Besonders schlecht sind die Ergebnisse in diesem worst-case-Szenario, wenn die Gesamtauslastung des Systems sehr hoch ist.

Hung und Leachman [HL99] gehen ähnlich vor wie Johnson et al., sie bilden aber eine Reihenfolge der Auslastungen, der durchschnittlichen Wartezeiten von Aufträgen vor Anlagen und der Standardabweichung der Wartezeiten. Es zeigt sich, dass eine hohe Korrelation besteht zwischen der Auslastung einer Anlage und der Standardabweichung der Wartezeiten vor dieser Anlage, je höher die Auslastung, umso höher die Streuung der Wartezeiten. In einer empirischen Studie wurden folgende Szenarien untersucht: Ein System mit 30 Anlagengruppen wurde in zwei Stufen vereinfacht. Auf der ersten Vereinfachungsstufe wurden die Anlagengruppen mit den 10 höchsten Streuungen der Wartezeiten bewahrt, auf der zweiten Stufe nur noch 5. Die Verweilzeiten der substituierten Anlagengruppen waren dabei in einem Szenario deterministisch in einem weiteren stochastisch verteilt. Alle Daten wurden in einer Testsimulation gewonnen. Die Tabellen 3.1 und 3.2 zeigen das Ergebnis der Untersuchungen.

Die Laufzeit konnte auf 19,6%, respektive 8,9% gesenkt werden. Dies entspricht ungefähr auch der Zahl der eingesparten Ereignisse; hier konnte die Anzahl auf 22%, respektive 9,8% gesenkt werden. Der Fehler in den Durchlaufzeiten blieb bei deterministischen

⁸M/M/10: Darstellung aus der Warteschlangentheorie (siehe z.B. [Kie97]). Ein System mit einer Warteschlange vor 10 parallelen Servern. Die Ankunftsintervallzeit und die Servicedauer ist exponentialverteilt.

Tabelle 3.2: Reduzierung der Laufzeit durch Vereinfachung [HL99]

Deterministisches Szenario	Vollständiges Modell	Reduziertes Modell I	Reduziertes Modell II
Anlagengruppen	30	10	5
Laufzeit [s]	2059,3	403,9	182,3
Ausgeführte Ereignisse	748377	164397	73376
Ausführdauer pro Ereignis [ms]	2,709	2,457	2,484

Verweilzeiten durchschnittlich unter dem Fehler bei stochastischen Verweilzeiten, obwohl diese für eine bessere Approximation der Streuung der Wartezeiten integriert wurden. Der Fehler und die Streuung der Durchlaufzeiten ist in dem 10-Anlagengruppen-Szenario mit weniger als 2% sehr gering. Allerdings konnte eine Verringerung der Anlagenauslastung festgestellt werden.

Jain et al. [JLGL99] führen eine ähnliche Studie wie Hung und Leachman durch, indem sie ebenfalls ein Modell einer Halbleiterfertigung mit denselben Methoden vereinfachen. Jain et al. erreichten allerdings wesentlich schlechtere Ergebnisse als Hung und Leachman. Besonders auffällig bei den Ergebnissen ist, dass die Fehler der Durchlaufzeiten abhängig von den einzelnen Produkten sind, die auf den Anlagen gefertigt wurden. Warum dies so ist, wird nicht weiter erläutert, auch fehlt eine Betrachtung der eingesparten Laufzeit durch die Vereinfachung.

Völker [Völ03, VG03] stellt ein Verfahren zur Vereinfachung von Simulationsmodellen vor, das zur simulationsbasierten Optimierung der Termin- und Kapazitätsplanung genutzt wird. Aus dieser Zielstellung ergeben sich einige Änderungen gegenüber den oben beschriebenen Ansätzen. Bei Völker werden keine Anlagen durch pauschale Verweilzeiten substituiert, sondern es werden Arbeitsvorgänge in Arbeitsplänen durch Übergangszeiten substituiert. Diese Übergangszeiten sind die Summe aus den Bearbeitungszeiten an Anlagen und den geschätzten Wartezeiten. Dadurch, dass Aufträge nicht mehr jede Anlage im Arbeitsplan belegen, sind die Auslastungen dieser geringer, was durch eine pauschale Kapazitätsreduzierung kompensiert werden muss. Es werden mehrere Indikatoren entwickelt, um zu substituierende Arbeitsgänge zu erkennen. Neben der Auslastung von Anlagen und der Streuung der Wartezeiten können Arbeitsgänge und Aufträge bspw. anhand der Auftragslänge oder dem Fertigstellungstermin substituiert werden. In einer empirischen Untersuchung stellt Völker fest, dass eine stufenlose Vereinfachung erzielt werden

konnte, die in einigen Parametrierungen⁹ des Algorithmus kleine Verhaltensabweichungen bei starken Vereinfachungsraten (bis 80%) erzielt. Auch bei von FIFO unterschiedlichen Prioritätsregeln konnten gute Ergebnisse erzielt werden. Diese vereinfachten Modelle wurden auf ihre Laufzeit untersucht und schon bei der Durchführung eines Simulationslaufs der Vereinfachung konnte ein Zeitgewinn gegenüber der Simulation des Ausgangsmodells festgestellt werden¹⁰. Bei der vom Autor getroffenen Bewertung der Ergebnisse in Bezug auf Validität muss allerdings beachtet werden, dass wegen der Ausrichtung dieses Verfahrens auf die Termin- und Kapazitätsplanung nur eine periodengenaue Einplanung der Aufträge angestrebt wurde. Zudem wird nicht erwähnt, inwieweit zufallsabhängiges Verhalten des Modells berücksichtigt wurde.

Die hier vorgestellten Verfahren zeigen, dass die Komplexität und die Laufzeit bei gleichzeitiger geringer Veränderung des Verhaltens um einen wesentlichen Betrag reduziert werden können. Die Verfahren von Rose, Johnson et al., Hung und Leachman und Jain et al. werden angewendet auf Modelle der Halbleiterindustrie, aber es gibt keinen Grund zu vermuten, dass die gewählten Methoden auf diese Modelle beschränkt sind, zudem auch Völker gute Ergebnisse mit einem auf diesen Methoden basierenden Verfahren erzielt. Die Ergebnisse von Rose und Jain et al. und Völker zeigen allerdings, dass eine Messung der Verhaltensabweichung des vereinfachten Modells nötig ist, da das Verhalten der Vereinfachung, u.U. aus unklarer Ursache, starke Abweichungen zum Ausgangsmodell zeigt. Bis auf Brooks und Tobias, die den Engpass analytisch bestimmen, bestimmen alle Verfahren den Engpass im System mittels der Analyse einer Simulation des Ausgangssystems. Eine analytische Bestimmung des Engpasses ist eleganter, aber u.U. von geringerer Qualität. Um eine qualitative Bewertung beider Verfahren abgeben zu können, müssten Versuche durchgeführt werden. Die Bewertung der Auswirkungen von unterschiedlichen Produktionsprogrammen ist nicht eindeutig. Während Hung und Leachman angeben, dass es nur kleine Abhängigkeiten gab zwischen unterschiedlichen Produktionsprogrammen und den Auslastungen der Anlagengruppen, so zeigen die Ergebnisse von Jain et al., dass der Produkt-Mix sehr wohl eine Rolle für die Validität der Vereinfachung spielen kann.

⁹Einige Parametrierungen zeigten aus nicht feststellbaren Gründen ein instabiles Vereinfachungsgrad-Verhaltensabweichung-Verhalten.

¹⁰Aus Gründen der Logik muss wohl angenommen werden, dass die Simulation zur Erfassung des Modellverhaltens (Bestimmung der Engstellen und der Wartezeiten) nicht mit in diese Rechnung aufgenommen wurde.

3.3.4 Einsatz von Simulationsmetamodellen

Ein Simulationsmetamodell, oder kurz Metamodell, ist ein Modell eines Modells. Dieses Metamodell soll ein mit dem Ausgangsmodell vergleichbares Verhalten aufweisen [HB00]. Bei der Erstellung eines Metamodells wird das Ausgangsmodell als Blackbox angenommen und nur dessen Ein- und Ausgangsverhalten aufgezeichnet, um daraus die Parametrierung des Metamodells abzuleiten [VB05]. Metamodelle sind in diesem Kontext keine ereignisdiskreten Simulationsmodelle, so dass ein Wechsel des Modellierungsformalismus stattfindet (vgl. Zeigler [ZPK00]).

Eine Simulation bildet eine Menge Eingangsparameter \vec{x} mit der Funktion $f(\vec{x}) = \vec{y}$ auf die Menge der Zielgrößen \vec{y} ab [HB00]. Jeder Eingabeparameter x_i besitzt einen eigenen Definitionsbereich D_i , genauso wie jede Zielgröße y_i ihren eigenen Wertebereich W_y besitzt. Die Definitions- und Wertebereiche können Zahlenmengen sein, aber auch endliche Symbolmengen (so können auch Bedienstrategien, Prioritätsregeln o.ä. Zielgrößen sein). Ein Metamodell ist dann eine Funktion $g(\vec{x})$ mit $f(\vec{x}) \approx g(\vec{x}) = \vec{\hat{y}}$. $\vec{\hat{y}}$ sind dabei die durch das Metamodell approximierten Zielgrößen.

Zur Vereinfachung von Simulationsmodellen kann ein Metamodell eingesetzt werden, weil es das Verhalten des Ausgangsmodells nachahmen und dabei eine bessere Laufzeit besitzen kann [Mer05, PCG00]. Die Exaktheit, mit der ein Metamodell das Verhalten eines Ausgangsmodells nachahmen kann, hängt von mehreren Faktoren ab. Die wichtigsten Faktoren sind die Art des verwendeten Metamodells, welche auf den Anwendungsfall angepasst sein muss, und die Qualität und der Umfang der Aufzeichnungen des Verhaltens des Ausgangsmodells.

Es existieren zahlreiche Arten von Metamodellen, die wichtigsten sind: Polynomielle Regression (PR), Response Surface Modelling (RSM), Multivariate adaptive Regression Splines (MARS), Radial Basis Functions (RBF), Kriging (GK) und Neuronale Netze (NN) [Bar98, JCS01, Bar92, VB05, PCG00]. Alle Arten haben ihre Vor- und Nachteile und optimalen Anwendungsgebiete. Jin et. al [JCS01] untersuchen in einer Studie mit mehreren Testszenarien die Arten PR, KG, MARS und RBF. Als Bewertungskriterien wurden die Genauigkeit, die Robustheit, die Effizienz, die Transparenz und die Einfachheit der Implementierung ausgewählt. Es zeigte sich, dass RBF in den meisten Szenarien die besten oder gleichwertige Ergebnisse liefert, insbesondere, wenn die Datenbasis an Testdaten klein ist und das Modell hochgradig nicht-lineares Verhalten aufweist. Völker [Völ03], Fishwick

[Fis89] und Panayiotou et al. [PCG00] untersuchen die Einsatzfähigkeit von Neuronalen Netzen, als eine Methode der Künstlichen Intelligenz, zur Metamodellierung. Bei Fishwick erreicht ein NN im Vergleich zu Linearer Regression und RSM schlechtere Werte in Bezug auf Genauigkeit. Völker und Panayiotou et al. attestieren NN im Gegensatz dazu gute Genauigkeit. Sie sehen die Hauptnachteile der NN in der aufwendigen und schwierigen Erstellung, da eine große Basis an aufgezeichneten Daten vorhanden sein muss und schnell eine Überanpassung an die Datenbasis erfolgen kann. In diesem Fall ist die Genauigkeit sehr hoch bei Eingabedaten, die auch im Training des Metamodells verwendet wurden, aber schlecht, wenn diese nicht verwendet werden. Ein besonderer Vorteil der NN ist ihre große Flexibilität. So können NN eine große Bandbreite an Definitions- und Wertebereiche der Ein- und Zielgrößen behandeln, wo die meisten anderen Verfahren auf numerische Werte beschränkt sind.

In dieser Arbeit sollen Teilmodelle vereinfacht werden, die Marken als Ein- und Zielgrößen verlangen, so dass ein Symbolraum als Wertebereich gewählt werden muss, daher sind nur die NN zur Metamodellierung anwendbar. Zudem soll eine stufenlose Vereinfachung möglich sein, d.h. es sollen Vereinfachungen erzeugt werden, die eine vorgegebene Komplexitätsreduzierung aufweisen. Dies ist mit Metamodellen nicht zu erreichen, weil die Komplexität eines Metamodells nicht unabhängig von Art und zu vereinfachendem Modell einstellbar ist.

3.4 Methoden zum Messen der Komplexität und Verhaltensabweichung

Wenn in einem Algorithmus die Komplexität eines Modells gezielt verändert werden soll, dann muss die Komplexität des Ausgangsmodells und des erzeugten vereinfachten Modells gemessen werden. Der Vergleich der beiden Werte erlaubt dann eine Aussage über die erzielte Komplexitätsreduzierung. Die Messung der Verhaltensabweichung ist zusätzlich erforderlich, weil sichergestellt werden muss, dass sich das vereinfachte Modell im Verhalten nur in einem tolerierbaren Rahmen vom Ausgangsmodell unterscheidet.

3.4.1 Messen der Komplexität eines Modells

Die Messung der Komplexität von Simulationsmodellen ist auch im Bereich der Modellvereinfachung kein übliches Vorgehen. In Arbeiten zur Vereinfachung von Simulationsmodellen (bspw. [HL99, BT00]) erfolgt die Bewertung der Vereinfachung durch den Vergleich der Laufzeiten von Ausgangsmodell und Vereinfachung. Ein absolutes Maß, das unabhängig vom verwendeten Rechner und dem Simulationswerkzeug¹¹ ist und bei dem keine Simulation nötig ist, wird nicht verwendet. Arbeiten, die sich mit der Erstellung eines solchen Maßes beschäftigen, sollen im Folgenden vorgestellt werden. Die folgenden Maße basieren darauf, dass das Simulationsmodell als Graph vorliegt. Schruben und Yücesan [SY93] verwenden dazu Simulationsgraphen¹², Wallace [Wal87] verwendet Action Cluster Incident Graphs, die den Simulationsgraphen recht ähnlich sind.

Wallace [Wal87] entwickelt ein Maß, das dem dynamischen Charakter der Simulation Rechnung tragen und nicht nur statisch bspw. die Anzahl Codezeilen¹³ messen soll. Wallace entwickelt dazu das Maß in Formel 3.4.

$$C_W = \sum_{i=1}^n (2 \cdot RW_i + 1,5 \cdot W_i + R_i) \cdot \frac{A_i}{N} \quad (3.4)$$

mit

C_W	Modellkomplexität nach Wallace
RW_i	Anzahl Variablen in Knoten i mit Schreib- und Lesezugriff
W_i	Anzahl Variablen in Knoten i mit alleinigem Schreibzugriff
R_i	Anzahl Variablen in Knoten i mit alleinigem Lesezugriff
A_i	Grad (Summe ein- und ausgehende Kanten) des Knoten i
N	Anzahl Knoten im Graphen

Diesem Maß liegt zugrunde, dass Variablenzugriffe, besonders Schreibzugriffe, aufwendig sind und einen wesentlichen Bestandteil der Berechnungsvorschriften der Ereignisse und damit der Zustandstransformation bilden (Transformationskomplexität). Durch den Faktor $\frac{A_i}{N}$ sollen Knoten stärker gewichtet werden, wenn der Grad des Knotens hoch ist. Je

¹¹Da die Modellbeschreibung in den meisten Fällen werkzeugspezifisch ist, ist eine Unabhängigkeit des Komplexitätsmaßes vom Simulationswerkzeug nur ideell möglich.

¹²Für detaillierte Beschreibungen von Simulationsgraphen siehe z.B. [Bus96, Bus95, IMW96, Sch83, SY88]

¹³Gebräuchliche Bezeichnung: Lines Of Code. Zu Zeiten von Wallace waren Simulationsmodelle im Allgemeinen eigenständige Programme; Simulationswerkzeuge mit grafischer Oberfläche waren nicht üblich. Ein Simulationsmodell wie ein Computerprogramm zu bewerten lag somit nahe.

höher der Grad eines Knotens, desto stärker ist die Verknüpfung mit anderen Knoten und desto stärker ist der Kontrollfluss (Kontrollkomplexität). Die Kontrollkomplexität wurde zur Erfassung der Dynamik der Simulation integriert. In einigen Tests mit mehreren Modellen stellt Wallace das Maß anderen Maßen zur Bestimmung der Programmkomplexität gegenüber und erhält undifferenzierbare Ergebnisse. Eine Gegenüberstellung mit der Laufzeit der Simulation der untersuchten Modelle erfolgt nicht.

Schruben und Yücesan [SY93] stellen vier verschiedene Komplexitätsmaße vor, das erste bestimmt rein statisch die Komplexität des Simulationsgraphen, die weiteren versuchen auch die Dynamik zu erfassen. Das Maß 3.5 misst ausschließlich die Anzahl Knoten, d.h. die Anzahl an Ereignissen.

$$\mathcal{C}_1 = |V(G)| \quad (3.5)$$

Maß 3.6 wurde so erweitert, dass es den dynamischen Eigenschaften einer Simulation Rechnung trägt. Es misst die Entscheidungsstruktur, d.h. wie viele Entscheidungen bzgl. des Nachfolgeknotens getroffen werden. Schruben und Yücesan begründen dieses Maß mit Untersuchungen, die belegen, dass die Komplexität von Simulationsmodellen alleine von der Entscheidungsstruktur abhängen. Dies entspricht der Einbeziehung der Kontrollkomplexität von Wallace.

$$\mathcal{C}_2 = \frac{|E(G)|}{|V(G)|} \quad (3.6)$$

$$\mathcal{C}_3 = \sum_{v \in V(G)} C(v) + \sum_{v, w \in V(G)} I(v, w) \quad (3.7)$$

In 3.7 ist $C(v)$ die Bewertung eines Ereignisses hinsichtlich Komplexität und $I(v, w)$ die Bewertung der Verknüpfung von v und w . Wie C und I gewählt oder gemessen werden ist nicht fest vorgegeben, für einen späteren Versuch wählen sie die Anzahl der Zustandsvariablen, die einem Ereignis zugeordnet sind, und den Grad des Ereignisses.

Im Bereich der Softwaremetriken wurde von McCabe [McC76] ein Verfahren vorgestellt, das anhand der Zahl unabhängiger Pfade durch den Graphen des Kontrollflusses eines Programms die Komplexität bestimmt. Schruben und Yücesan wenden dieses Maß 3.8 auf Simulationsgraphen an¹⁴.

$$\mathcal{C}_4 = |E(G)| - |V(G)| + 1 \quad (3.8)$$

¹⁴Die Bestimmung der Cycomatic Number nach Formel 3.8 setzt voraus, dass der Graph aus nur einer Zusammenhangskomponente besteht und stark zusammenhängend ist. Der Modellgraph musste somit vor Anwendung des Maßes durch das Einfügen von miteinander verbundenen Supersenke und -quelle leicht modifiziert werden.

In einer Testphase mit mehreren unterschiedlichen Modellen fanden Schruben und Yücesan heraus, dass C_3 die größte Korrelation mit der Laufzeit der Simulation hatte. Demnach hat die Komplexität eines Ereignisses hohen Einfluss auf die Laufzeit, aber der Einfluss der Kontrollkomplexität, also die Verzweigung des Graphen als sekundäre Einflussquelle, muss ebenfalls mit berücksichtigt werden.

Das Maß 3.7 scheint geeignet zu sein, auch den Einfluss der Dynamik auf die Komplexität eines Simulationsmodells zu erfassen. Durch die genauere Ermittlung der Komplexität von Ereignissen $C(v)$ könnte aber ein besseres Komplexitätsmaß erzielt werden. Eine Untersuchung erscheint interessant, ob eine Verbesserung erzielt werden kann, wenn anstelle des Grades die Anzahl der Zyklen, auf denen ein Knoten liegt, für $I(v, w)$ verwendet wird.

3.4.2 Messen des Verhaltens und Validieren eines Modells

Das Verhalten eines Modells kann während der Simulation durch Bildung und Verfolgung von Kennwerten bestimmt werden. Zur Analyse von Produktionssystemen, die alleiniger Untersuchungsgegenstand der hier behandelten Materialflusssimulation sind, können u.a. Produktivitätskennzahlen aus dem Produktionscontrolling bestimmt werden [HH01]. Kennzahlen des mengen- und zeitorientierten Inputcontrollings sind bspw. arbeitsbezogene Kennzahlen ($\frac{\text{Tätigkeitszeit}}{\text{Arbeitsstunden}}$, etc.), materialbezogene Kennzahlen (Vorratsreichweite) und betriebsmittelbezogene Kennzahlen (Maschinenstillstandszeit, Störquote). Im mengen- und zeitorientierten Throughputcontrolling können Kennzahlen wie der Leistungsgrad berechnet werden. Zur Bestimmung des Umlaufbestands und der Durchlaufzeit bietet sich der Ansatz der Fortschrittszahlen an, der eine kumulierte Mengengröße darstellt, die auf einen bestimmten Artikel und einen Zeitpunkt bezogen wird [GG92]. Im Konzept der Fortschrittszahlen können Kontrollblöcke definiert werden, die für die Bewertung hierarchischer Zusammenfassungen von Betriebsmitteln (Abteilung oder Bereich) genutzt werden können. Dies weist eine Analogie zu den Teilmodellen dieser Arbeit auf. Im Rahmen der PPS können Fortschrittszahlen und Kontrollblöcke also nur effektiv in der Fließfertigung eingesetzt werden, wenn der Materialfluss nur in einer Richtung durch den Kontrollblock fließt und das Produktspektrum und die Absatzsituation die Eigenschaften der (Groß-)Serien- oder Massenfertigung aufweist. Von dieser Voraussetzung für den Einsatz der Fortschrittszahlen kann in dieser Arbeit abgewichen werden, weil nur die Erfassung des Verhaltens und keine Steuerung des Produktionssystems erfolgen soll.

Werden nun für ein Realsystem, oder im Falle der Vereinfachung für das Ausgangsmodell, und für ein zu validierendes Modell diese Kennzahlen gebildet, stellt sich die Frage, wie effektiv überprüft werden kann, wie stark etwaige auftretende Abweichungen sind. Um diese Frage zu beantworten, bieten sich unterschiedliche Verfahren der Validierung an, denen eine mehr oder weniger starke statistische Formalisierung zugrunde liegt [Sar05]. Viele Techniken sind nur manuell mit Hilfe von Expertenwissen ausführbar (Animation, Vergleich von Diagrammen, etc.). Die Verwendung von Konfidenzintervallen und Hypothesentests wird von Sargent [Sar05] für einen Anwendungsfall empfohlen, wenn die Systeme beobachtbar sind und objektive Ergebnisse erzielt werden müssen. Im Falle der Modellvereinfachung, wenn also beide Vergleichsobjekte vorhanden sind, ist eine Möglichkeit, wie Kleijnen [Kle99], „trace-driven“ zu validieren. Bei der „trace-driven“ Validierung erhalten das Referenzsystem und das zu untersuchende Modell beide dieselbe Trajektorie an Eingangsdaten¹⁵ und die Trajektorien der Ausgangsdaten werden verglichen. Ein Vergleich der Trajektorien erfolgt dann mit statistischen Mitteln wie Mittelwert- und Standardabweichungsbestimmung oder Hypothesentests. Einen automatisierten Ansatz der „trace-driven“ Validierung durch Nutzen von Methoden der künstlichen Intelligenz stellen Martens et al. [MPK06] vor.

Die Erzeugung von Kennzahlen des Produktionscontrollings in Kombination mit einem „trace-driven“ Ansatz zur Bestimmung des Verhaltensunterschieds zwischen Ausgangsmodell und Vereinfachung erscheint für diese Arbeit geeignet. Es ist im Verlauf dieser Arbeit zu entscheiden, welche Kennzahlen gebildet werden sollen und wie die aufgezeichneten Trajektorien verglichen werden sollen.

3.5 Methoden zum Erzeugen und Verändern von Zuständen

3.5.1 Zustände im Bereich Parallele und Verteilte Simulation

Eine große Bedeutung hat die Betrachtung von Zuständen und die Bewahrung ihrer Konsistenz in der Parallelen und Verteilten Simulation [Fuj98]. Dort wird die Menge der im Modell befindlichen Komponenten in Teile unterteilt und diese auf unterschiedlichen

¹⁵Wird gegen ein Realsystem validiert, erhält das zu untersuchende Modell die Eingangsdaten, die im Realmodell aufgezeichnet werden konnten.

Prozessoren gleichzeitig berechnet. Zur Berechnung von parallelen Simulationen existieren konservative und optimistische Verfahren. Die konservativen verhindern jede Form von Inkonsistenz zwischen den Prozessoren, erreichen aber nur eine geringe Laufzeitverbesserung. Die optimistischen Verfahren erlauben das Entstehen von Inkonsistenzen, es existieren aber Mechanismen, diese zu entdecken und zu beheben. Es wartet nicht jeder Prozessor mit der Berechnung seiner Komponenten auf Rückantwort von anderen Prozessoren, die Nachfolger von seinen Komponenten berechnen. Dadurch können Inkonsistenzen entstehen. Wird eine solche erkannt, so springt der Prozessor zu einem früheren, abgespeicherten Zustand zurück und berechnet seinen Zustand mit den Eingaben seines Vorgängers bis zum Zeitpunkt des Erkennens der Inkonsistenz neu. Ändert sich während dieser Neuberechnung das Ausgangsverhalten des Prozessors, so werden der oder die Nachfolger ebenfalls neu berechnet. Dies kann dazu führen, dass im schlimmsten Fall das gesamte Modell erneut berechnet werden muss. Die Laufzeit der parallelen Simulation wird durch das häufige Abspeichern von Zuständen und das erneute Berechnen von Modellteilen negativ beeinflusst. Es existieren viele Arbeiten, die sich mit Techniken zum effizienten Abspeichern und Neuberechnen von Zuständen beschäftigen (engl. Begriff: Time Warp Parallel Simulation). Diese benutzen allerdings häufig Annahmen, die im Bereich der dvdS nicht zutreffen, wie einen kleinen Zustandsraum oder sich wiederholende gleiche Zustände [SR96, LT04].

In einem anderen Ansatz werden nicht die Komponenten in Teile unterteilt, sondern der Simulationszeitraum [KP04]. Auf jedem verwendeten Prozessor werden hier alle Komponenten berechnet, aber nur in einem kleinen Zeitintervall. Da alle Zeitintervalle gleichzeitig berechnet werden, müssen die Zustände für den Beginn eines jeden Intervalls geraten werden. Ist der Endzustand eines Intervalls nach Simulation nicht gleich dem geratenen Anfangszustand des nächsten Intervalls, so muss dieses Zeitintervall mit neuem Startzustand erneut berechnet werden. Auch dieses Verfahren ist in der dvdS nicht anwendbar, denn könnte ein Zustand der Simulation genau geraten werden, so wäre eine Simulation nahezu überflüssig.

Das nachfolgend beschriebene Verfahren der Nachsimulation von Mueck [Mue05] (siehe Abschnitt 3.5.3) lehnt sich sehr stark an die Problematik und die Methodik der Verfahren in der parallelen Simulation an. Wie aus dem Zustand eines Modell der Zustand eines ähnlichen Modells berechnet werden kann, ohne eine Simulation zu verwenden, wird in der Literatur nicht beschrieben, was höchstwahrscheinlich daran liegt, dass dieses Problem vor der dvdS noch nicht aufgetaucht ist.

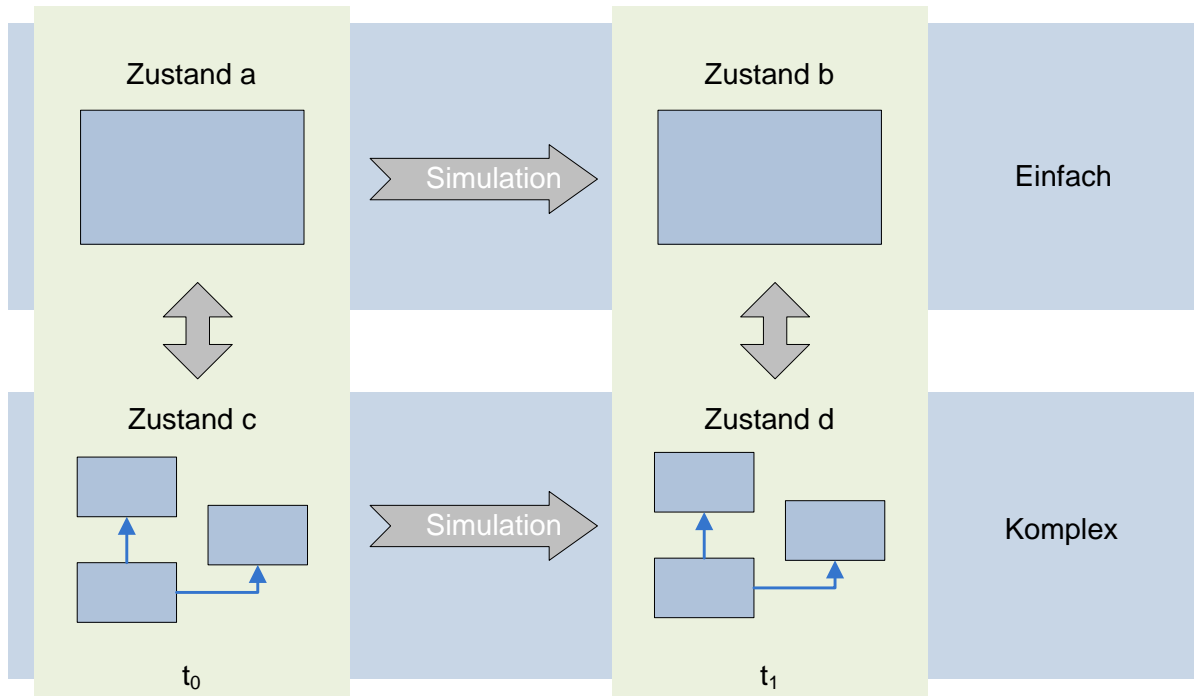


Abbildung 3.3: Konsistenzanforderungen an die Umschaltvorgänge

3.5.2 Anforderungen an die Konsistenz beim Austausch von Teilmodellen

Soll ein Teilmodell in einer dvdS aktiviert werden, so muss es einen Zustand aufweisen, der dem des deaktivierten weitestgehend entspricht, damit die Konsistenz der Simulation gewahrt bleibt. Es wird hierfür ein Zustand des aktivierten Teilmodells aus dem Zustand des deaktivierten berechnet werden.

Davis [Dav92] und Palmore [Pal92] definieren im Rahmen des multi-resolution modeling Konsistenzanforderungen an Umschaltvorgänge. Die Anforderungen sollen anhand Abbildung 3.3 erläutert werden. Davis und Palmore fordern, dass zum Zeitpunkt t_1 folgendes gilt:

Nr.	Zustandsgleichheit zum Zeitpunkt t_1	Pfade zum Erreichen der Zustände
1.	$d_\alpha = d_\beta$ wenn	$a_\alpha \rightarrow c_\alpha \rightarrow d_\alpha$ und $a_\beta \rightarrow b_\beta \rightarrow d_\beta$
2.	$b_\alpha = b_\beta$ wenn	$a_\alpha \rightarrow b_\alpha$ und $a_\beta \rightarrow c_\beta \rightarrow d_\beta \rightarrow b_\beta$
3.	$b_\alpha = b_\beta$ wenn	$c_\alpha \rightarrow a_\alpha \rightarrow b_\alpha$ und $c_\beta \rightarrow d_\beta \rightarrow b_\beta$
4.	$d_\alpha = d_\beta$ wenn	$c_\alpha \rightarrow a_\alpha \rightarrow b_\alpha \rightarrow d_\alpha$ und $c_\beta \rightarrow d_\beta$

Alle hier vorgestellten Konsistenzanforderungen von Davis und Palmore werden als wenig realistisch angesehen, weil die Verhaltensunterschiede zwischen einem komplexen und einem vereinfachten Modell, um einen gleichen Zustand generieren zu können, so gering sein müssten, dass die Verwendung unterschiedlich komplexer Modelle unzweckmäßig wäre. Würde eine Zustandsabbildungsfunktion existieren, die aus einem wesentlich vereinfachten Modell den Zustand eines komplexen Modells errechnen könnte, so wäre der Einsatz des komplexen Modells sinnlos. Wesentlich realistischer und für diese Arbeit anwendbar wäre eine Definition, die zur Festlegung einer Toleranz ein ϵ vorsieht (z.B.: $d_\alpha = d_\beta + \epsilon$).

3.5.3 Erzeugung von konsistenten Zuständen in der dvdS

Als Berechnungsfunktionen für konsistente Zustände schlägt Mueck [Mue05] folgende Verfahren vor :

- Nachsimulation

Ein Zustand wird erzeugt, indem die Simulationszeit seit der letzten Aktivierung erneut für ein Teilmodell simuliert wird. Dabei wird das Teilmodell als freigeschnitten betrachtet. Es erhält die „verpassten“ Ausgaben seiner Vorgänger als Eingaben¹⁶, hat aber keinen Einfluss auf seine Nachfolger. Aus diesem Grund können Inkonsistenzen auftreten und es ist nicht gewährleistet, dass das zu aktivierende Teilmodell den Zustand einnimmt, den es hätte, wenn es ständig aktiv gewesen wäre.

- Nachsimulation mit Verfolgung der Interaktionen

Um den Fehler der Nachsimulation zu reduzieren, werden bei diesem Verfahren auch Teilmodelle mit berechnet, die mit dem zu aktivierenden verknüpft sind. Dadurch wächst der Berechnungsaufwand, da mehr Teilmodelle berechnet werden müssen.

- Direkte Berechnung

Dieser Methode liegt die Annahme zu Grunde, dass sich aus dem Zustand des deaktivierten Teilmodells funktional der Zustand des aktivierten berechnen lässt. Dieser funktionale Zusammenhang muss a priori für die Teilmodelle festgelegt werden. Hierfür sind Prozesswissen des Modellierers und Informationen über das Vorgehen in der Vereinfachung vonnöten.

¹⁶Also die Eingaben, die aktive Bausteine erhalten haben, deren Stelle das zu aktivierende einnehmen soll. War das zu aktivierende Teilmodell schon einmal aktiv, dann erhält es die Eingaben seit seiner Deaktivierung.

Tabelle 3.3: Übersicht Übergangsfunktionen [Mue05]

Methode	Rechenaufwand	Modellierungsaufwand	Konsistenzprüfung nötig?	Genauigkeit
Nachsimation	–	++	Ja	+
Nachsimation mit Verfolgung der Interaktionen	– –	++	Nein	++
Direkte Berechnung	–/+	– –	Optional	+

In Tabelle 3.3 werden die Eigenschaften der oben genannten Verfahren nochmals bewertend zusammengefasst. Die Bewertung des Berechnungsaufwands in Tabelle 3.3 ist abhängig von der Simulationszeit, für die ein Teilmodell inaktiv war. Je länger diese Zeitspanne ist, umso länger muss das Teilmodell simuliert werden. Da keine Aussage darüber getroffen wurde, wie aufwendig die direkte Berechnung ist, kann keine Aussage über Vor- und Nachteile gemacht werden und der Wert in Tabelle 3.3 kann als „unbestimmt“ interpretiert werden. Die Notwendigkeit einer Konsistenzprüfung ist nur optional, wenn die Zustandsabbildungsfunktion stabil ist, also in allen Situationen gute Zustände berechnet. Ist sie es nicht, dann ist eine Konsistenzprüfung zwingend erforderlich. Mueck trifft hierzu keine Aussagen.

3.6 Die Simulationsplattform

Die Simulationsplattform ist der ereignisdiskrete Simulator d³FACT [DMMH05, DHLM05, MDL⁺04, DFH⁺08, SFH⁺08], der von den Fachgruppen Algorithmen und Komplexität (Prof. Meyer auf der Heide) und Wirtschaftsinformatik, insb. CIM (Prof. Dangelmaier) des Heinz Nixdorf Instituts mit Unterstützung der DFG¹⁷ entwickelt wird. Wesentliche Module von d³FACT wurden von Mueck und Laroque im Rahmen ihrer Dissertationen [Mue05, Lar07] entwickelt. Im Folgenden wird zuerst auf spezielle Module von Mueck und Laroque eingegangen, die für diese Arbeit wichtig sind, danach erfolgt ein allgemeiner Überblick über das Softwareprojekt.

¹⁷Deutsche Forschungsgesellschaft: Forschungsprojekte BAMSI und AVIPASIA

3.6.1 Benutzerstimulierte detaillierungsvariante Simulation

Mueck [Mue05, DM04] konzipierte in seiner Dissertation ein Verfahren zur benutzerstimulierten detaillierungsvarianten Simulation von Materialflüssen. Besondere Eigenschaft dieser Methode ist, dass, eingebettet in eine Virtual-Reality (VR) Visualisierung, der analysierende Benutzer die Komplexitätsgrade der verwendeten Teilmodelle definiert. In der Nähe des Benutzers befindliche Teilmodelle und solche, die mit diesen logistisch eng verknüpft sind, sind hoch komplex, die übrigen in mehreren Abstufungen weniger komplex. Bewegt sich der Benutzer, so wird durch das Aktivieren und Deaktivieren (auch Umschalten) von Teilmodellen erreicht, dass die Zonen der Komplexitätsabstufung auf den Untersuchungsfokus des Benutzers zentriert bleiben.

Das Umschalten von Teilmodellen erfolgt so, dass die Verknüpfungen von weiterhin aktiven Teilmodellen zu den zu deaktivierenden Teilmodellen auf die zu aktivierenden „umgebogen“ werden. Die deaktivierten Teilmodelle liegen also nicht mehr im Markenstrom und Ereignisse werden nicht mehr betrachtet. Wann welches Teilmodell aktiv ist, wird von mehreren Indikatoren bestimmt, mit denen bewertet wird, auf welcher Komplexitätsstufe die Bereiche des modellierten Systems simuliert werden sollen. Mueck entwickelt zur Bewertung der Signifikanz eines Bereichs vier Indikatoren, deren Grundannahme es ist, dass der Benutzer seinen visuellen Fokus auf den Untersuchungsgegenstand legt:

- Abstand
Um die Position des Benutzers in der VR werden Kreise unterschiedlichen Radius gelegt. Für im kleinsten Kreis gelegene Komponenten sollen Teilmodelle höchster Komplexität verwendet werden. Zwischen diesem ersten und dem zweiten Kreis sollen die Teilmodelle von der nächstgeringeren Komplexitätsstufe sein, usw. für alle definierten Komplexitätsstufen.
- Sichtfeld
Komponenten, die nicht im Sichtkegel des Benutzers liegen, sollen in einer geringeren Komplexität berechnet werden als solche im Sichtkegel.
- Verdeckung
Komponenten, die im Sichtfeld liegen, aber von größeren oder näher an der Benutzerposition befindlichen Objekten verdeckt werden, sollen in einer geringeren Komplexität berechnet werden.

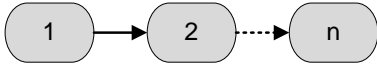
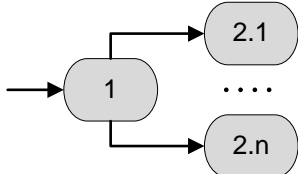
- Logistische Verkettung

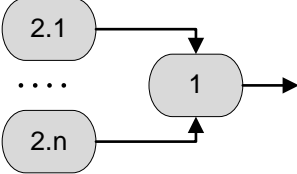
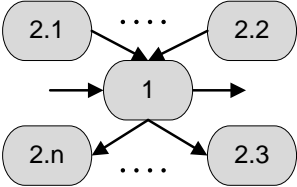
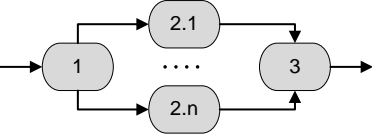
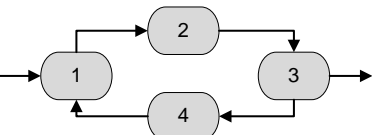
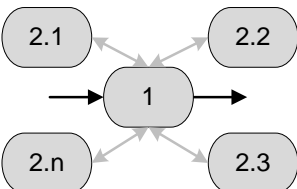
Die mit den Komponenten im Untersuchungsfokus logistisch verknüpften Komponenten, also die im Materialfluss vor- oder nachgelagerten Komponenten oder Steuerungen, haben großen Einfluss auf das Verhalten der untersuchten Komponente. Diese logistisch verketteten Komponenten sollen mit höherer Genauigkeit berechnet werden. Die Verfolgung der Verknüpfung erfolgt nur bis zu einer bestimmten Tiefe, da u.U. bei einem stark zusammenhängenden Modell alle Komponenten mit der Komponente im Untersuchungsfokus indirekt verkettet sind.

3.6.2 Rückwärtssimulation

Die Fähigkeit zur Rückwärtssimulation wurde von Laroque [Lar07] in d³FACT integriert, um eine Planung analog zur Rückwärtsplanung in der PPS zu ermöglichen. Es wurde eine Methode entwickelt, die eine teilautomatisierte Transformation von vorwärtsgerichteten Modellen in rückwärtsgerichtete erlaubt. Für diese Transformation werden iterativ Strukturen von Komponenten, ähnlich dem Vergrößerungsschritt der Multilevel-Partitionierung, zu einzelnen Metakomponenten zusammengefasst. Laroque identifiziert nach [FHD⁺00] eine Liste von Materialfluss-Grundstrukturen (siehe Tabelle 3.4), die in der Lage sind, die überwiegende Anzahl aller praxisrelevanten Materialflüsse abzubilden.

Tabelle 3.4: Übersicht Grundstrukturen

Quelle und Senke		Quellknoten erzeugen Marken und haben keinen Vorgänger. Senken terminieren den Materialfluss und haben keinen Nachfolger.
Unverzweigte Linie		Unidirektional gekoppelte Komponenten, von denen jede nur einen Vorgänger und einen Nachfolger besitzt.
Verzweigung		Eine Komponente besitzt einen Vorgänger und mehrere Nachfolger

Zusammenführung		Eine Komponente besitzt mehrere Vorgänger und einen Nachfolger
Kreuzung		Kombination aus Verzweigung und Zusammenführung und besitzt beliebige Zahl (>1) an Vorgängern und Nachfolgern.
Parallele Linie		Kombination aus einer Verzweigung, einer Zusammenführung und mehreren unverzweigten Linien. Der Materialfluss zwischen Verzweigung und Zusammenführung muss ein geschlossenes System bilden.
Rückkopplung		Kombination aus einer Verzweigung und einer Zusammenführung und zwei unverzweigten Linien, bei der die Materialflussrichtung der beiden Linien entgegengesetzt verläuft.
Stern		Eine zentrale Komponente ist bidirektional mit mehreren Komponenten verbunden. Der Materialfluss kann dieses System nur über die zentrale Komponente betreten oder verlassen.

Sind alle Komponenten und Metakomponenten vorhergegangener Iterationen in einer einzigen Metakomponente zusammengefasst, kann das Modell rekursiv transformiert werden, indem der Inhalt jeder Metakomponente nach einer der Struktur entsprechenden Regel transformiert wird. In Abbildung 3.4 wird beispielhaft ein Iterationsschritt in dieser Zusammenfassungsmethode gezeigt, bei dem die dunkel hinterlegten Knoten als Grundstrukturen erkannt und zu einem Knoten zusammengefasst werden. Die Rückwärtssimulation

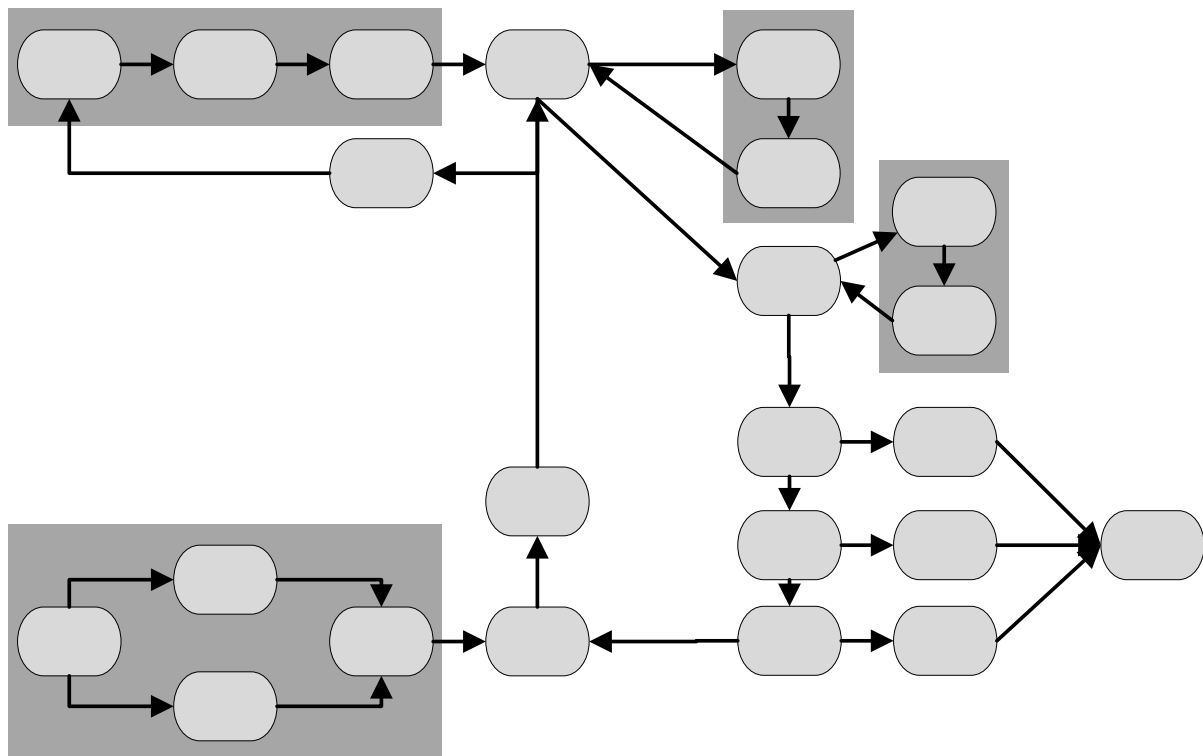


Abbildung 3.4: Strukturerkennung zur iterativen Zusammenfassung.

wird in dieser Arbeit nicht verwendet, aber die Definition der Grundstrukturen und die Modellanalyse zum Finden dieser werden in die Vereinfachung einfließen.

3.6.3 Sonstige Module von d³FACT

In diesem Abschnitt soll ein kurzer Überblick über die Funktionsweise des Simulators gegeben werden: die Implementierung der Modellierungsmethode, ein Modul zur Modellierung, die immersive 3D Visualisierung und die Implementierung der dvdS-Funktionalität.

Der Simulationskernel

Der Simulationskernel führt die Simulation eines Modells aus. Er beinhaltet die Simulationsuhr, die Ereignisliste und Methoden zum Verändern des Simulationszustands, die von den Ereignisroutinen aufgerufen werden. Um weitere Module, wie bspw. eine Visualisierung, anschließen zu können, besitzt er eine Nachrichtenschnittstelle, über die der

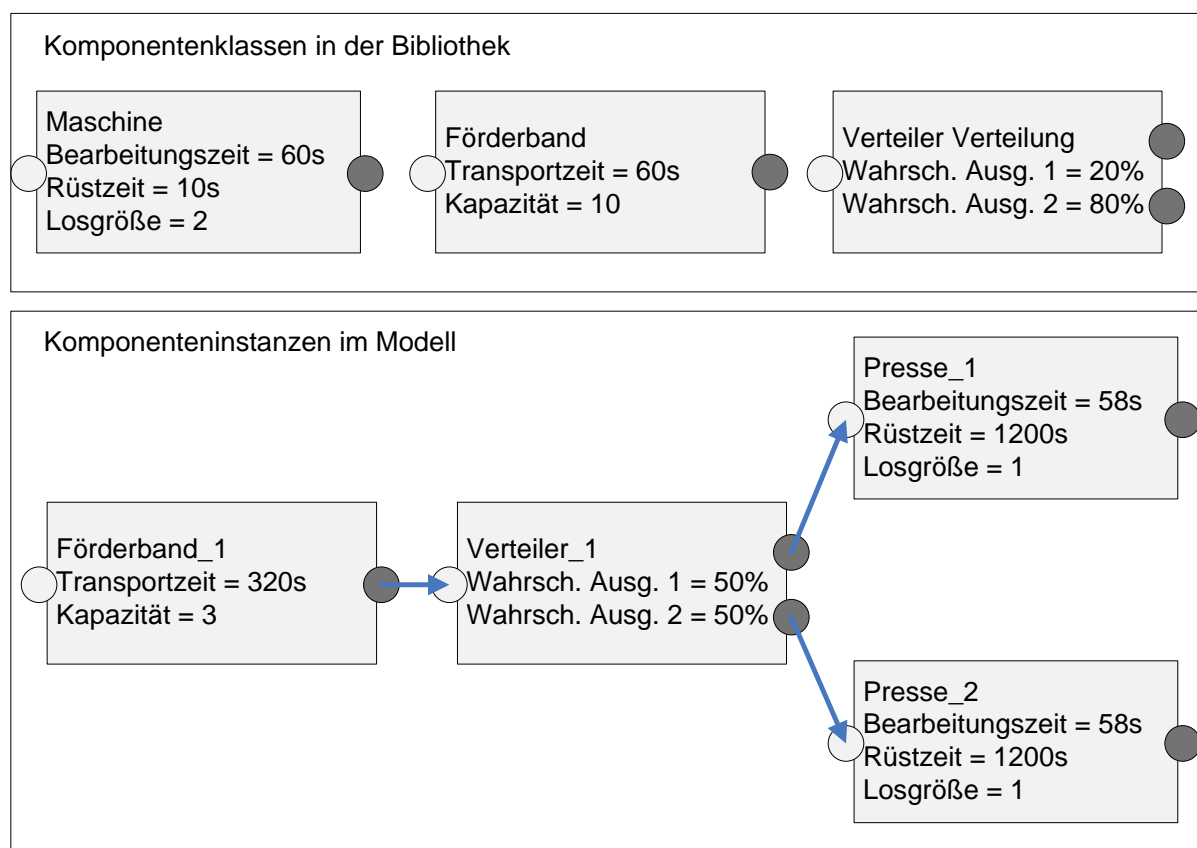


Abbildung 3.5: d³FACT Modellierungsskizze

Simulationszustand nach außen kommuniziert wird und zustandsverändernde Eingaben entgegengenommen werden.

Implementierung der Modellierungsmethode

Die Modellierungsmethode (vgl. 3.1.3) wurde so umgesetzt, dass zur einfachen Wiederverwendung Komponentenklassen erstellt werden, die dann in einem konkreten Modell instantiiert, verknüpft und parametrisiert werden (siehe Abbildung 3.5). Modelle werden in einer XML¹⁸-Struktur abgelegt, wie sie in Programmauflistung 3.1 skizziert ist. Die Struktur dieser XML-Datei wird über eine DTD¹⁹ festgelegt und kann so automatisch auf strukturelle Validität geprüft werden. Wenn ein Modell simuliert werden soll, dann wird diese XML-Datei mittels einer XSL²⁰-Transformation in Java-Klassen umgesetzt. Diese

¹⁸eXtensible Markup Language

¹⁹Document Type Definition

²⁰eXtensible Stylesheet Language

Java-Klassen bilden dann ein ausführbares Java-Programm, die Simulation.

Programmauflistung 3.1: Vereinfachte Darstellung des XML-Schemas

```
<model>
  <library>
    <componentclass name="A">
      <component name = "B_1" type="B">
        <position/>
        <variable_value/>
      </component>
      <component name = "B_2" type="B">
        <position/>
        <variable_value/>
      </component>
      <link> </link>
    </componentclass>
    <componentclass name="B" mesh="1">
      <position/>
      <ichannel> </ichannel>
      <ochannel> </ochannel>
      <variable> </variable>
      <event> </event>
    </componentclass>
  </library>
  <modeltorun> A </modeltorun>
</model>
```

Ereignisroutinen

Ebenfalls in der Modell-XML-Datei werden die Ereignisroutinen definiert. Dies erfolgt in einer angepassten Java-Syntax, wie in Programmauflistung 3.2 zu sehen ist²¹.

Programmauflistung 3.2: Beispiel für Eventcode

```
try
{
```

²¹Die dargestellte Ereignisroutine entnimmt eine Marke (Token) aus einem Speicher und versendet sie über den Channel mit dem Namen oc1. Sollte dies erfolgreich sein, dann wird der Input-Channel wieder geöffnet, um neue Marken entgegenzunehmen, andernfalls wird eine Informationsmeldung erzeugt.


```
myEntity.getOutputStream("oc1").add((Token) myEntity. ↘  
    getArrayListVar("TokenArray").getFirst());  
myEntity.getArrayListVar("TokenArray").removeFirst();  
myEntity.getInputChannel("ic1").open();  
}  
catch (ClosedException e ) {Logger.info(myEntity,"Versenden ↘  
    fehlgeschlagen.");}
```

Ereignisroutinen werden in 8 verschiedene Ereignistypen unterteilt, die festlegen, wann eine Routine ausgeführt wird oder wie der Simulationszustand mit ihr verändert werden soll.

- **Init-Ereignis**
Erzeugen des initialen Zustands, wird zum Zeitpunkt t_0 ausgeführt.
- **Input-Ereignis**
Ist einem Inputchannel zugeordnet und wird ausgeführt, wenn eine Marke in die Komponente über den Channel eintritt.
- **Output-Ereignis**
Ist einem Outputchannel zugeordnet und wird zum Verschicken einer Marke über den Channel ausgeführt.
- **Reopen-Ereignis**
Ist einem Outputchannel zugeordnet und wird ausgeführt, wenn der Inputchannel, der mit dem Outputchannel verbunden ist, seinen Status von geschlossen auf offen schaltet, allerdings nur, falls vorher ein Output-Ereignis daran gescheitert ist, eine Marke zu versenden, weil der verbundene Inputchannel geschlossen war.
- **UserDefined-Ereignis**
Dieses Ereignis kann von anderen Ereignissen aufgerufen werden und frei definierbare Funktionen erfüllen.
- **Switch-Ereignis**
Wird bei einer Umschaltung zur Zustandsgenerierung ausgeführt.
- **Final-Ereignis**
Wird ausgeführt, wenn das Ende der Simulation erreicht ist, das kann das Erreichen einer bestimmten Simulationszeit oder eines Zustands sein.

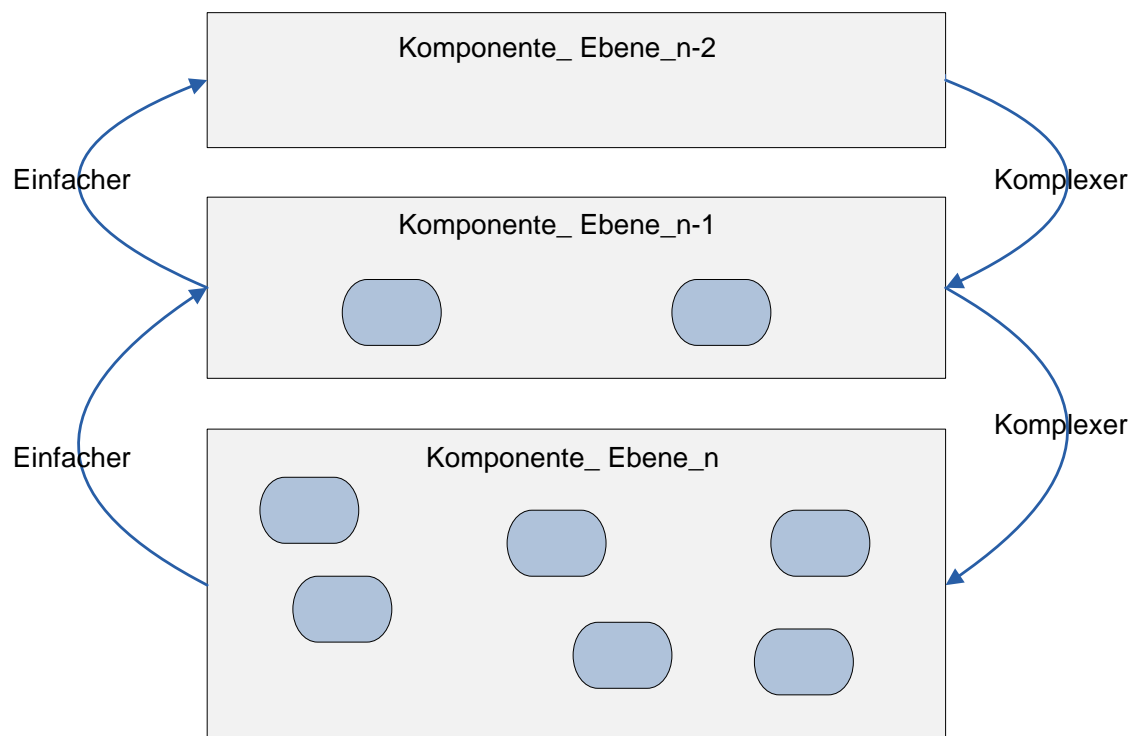


Abbildung 3.6: Modellstruktur für die detaillierungsvariante Simulation

Implementierung der dvdS-Funktionalität

Die von Mueck [Mue05] konzipierte dvdS wurde für d³FACT adaptiert und teilweise integriert. Jeder Komponentenklasse kann eine Komponentenklasse zugeordnet werden, die ein komplexeres Modell derselben darstellt, und eine weitere Komponentenklasse als weniger komplexer Repräsentant (siehe Abbildung 3.6). Hat eine Komponente Repräsentanten in anderen Komplexitätsebenen, so müssen Umschaltereignisse (Switch-Ereignis, s.o.) vorgesehen werden. In diesen Ereignissen (beispielhaft in Programmauflistung 3.3) werden die Verknüpfungen des Modells angepasst, also die deaktivierte Komponente wird ausgekoppelt, und der Zustand der aktivierten Komponente wird berechnet. Diese Berechnung entspricht der direkten Berechnung des Zustands nach Mueck (siehe Abschnitt 3.5).

Programmauflistung 3.3: Beispiel für eine Ereignisroutine zum Umschalten

```
assert myLogger.info(myEntity, "Switch Event Gamma In");
setChannels(myEntity.in);
// Token :
if (myEntity.getinstance_beta1().gettoken() != null)
{
    ((alpha)myEntity.in).settoken(myEntity.getinstance_beta1().gettoken());
}
```

```
        gettoken() );
    }
    else if (myEntity.getInstance_beta2().gettoken() != null)
    {
        ((alpha)myEntity.in).settoken(myEntity.getInstance_beta2().gettoken() );
    }
    // Events :
    HashMap map = new HashMap();
    map.put(new UserDefinedEvent_beta_endOfDelay(), new \
        UserDefinedEvent_alpha_endOfDelay((alpha)myEntity.in));
    kernel.replaceEvents(myEntity.getInstance_beta1(), map);
    kernel.replaceEvents(myEntity.getInstance_beta2(), map);
```

Werkzeug zur Modellierung

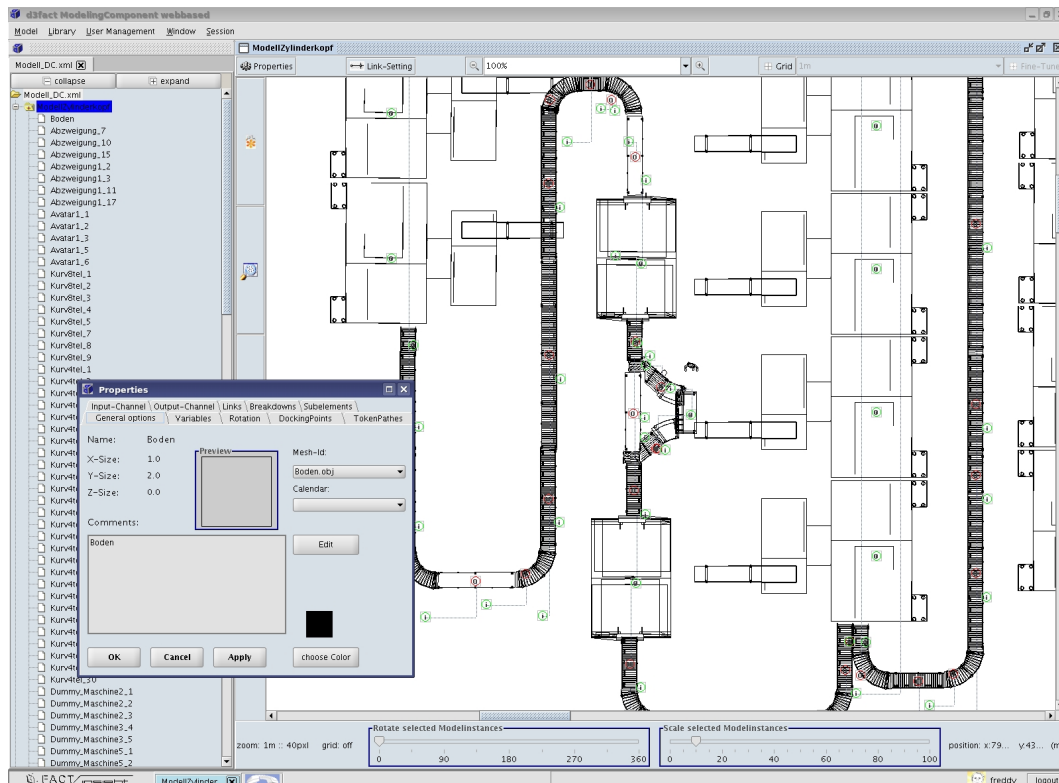
Zur benutzerfreundlichen Modellierung wurde ein Modellierungswerkzeug erstellt, mit dem d³FACT Modelle mit einer grafischen Oberfläche erstellt werden können. Als Alternative dazu kann die XML-Datei minimalistisch in einem Texteditor bearbeitet werden. Zur effektiven Modellierung großer Systeme ist das Modellierungswerkzeug als eine mehrbenutzerfähige Client-Server-Anwendung [DAL⁺06] (siehe Abbildung 3.7(a)) konzipiert worden, in der Benutzer in kooperativer Arbeit gemeinsam ein Modell erstellen können.

Die immersive 3D Visualisierung

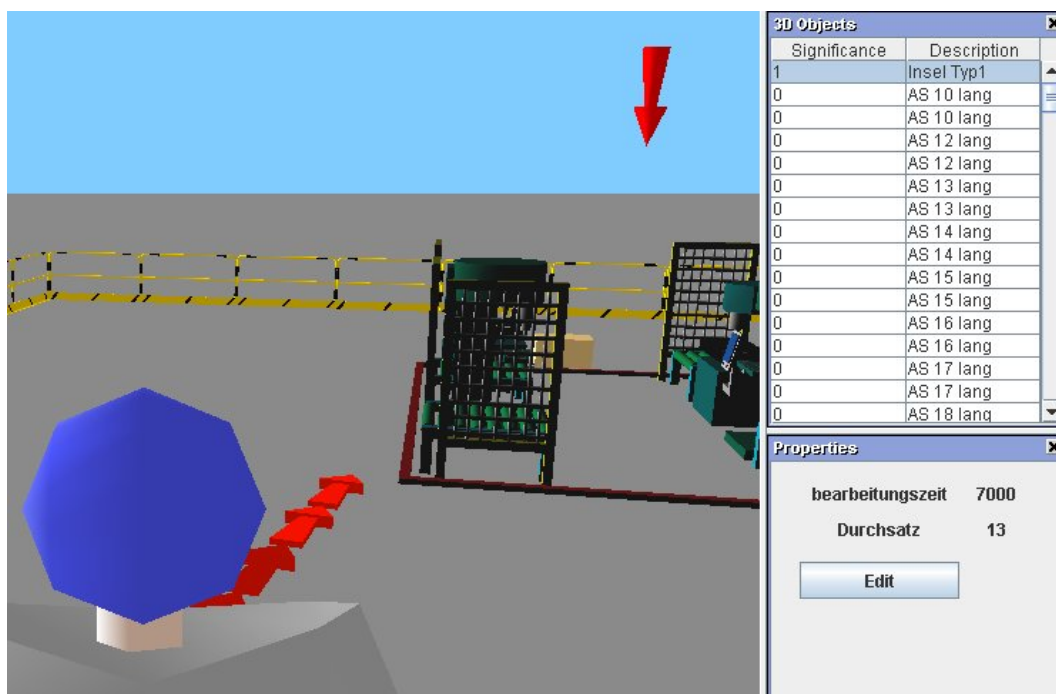
Eine Simulation kann in einer mehrbenutzerfähigen virtuellen Realität analysiert und beeinflusst werden. Jeder Benutzer, der die Simulation „betreten“ hat und sie untersuchen will, erhält einen Avatar²², welcher eine definierte Position auf der x-y-Ebene²³ und einen Sichtbereich hat (siehe Abbildung 3.7(b)). Parameter von Komponenten können geändert werden, um das Verhalten der Komponenten den Wünschen des Benutzers anzupassen. Zur effektiven Multi-User-Unterstützung können Benutzer mittels Textnachrichten oder IP-Telefonie kommunizieren.

²²Stellvertreter einer Person in der virtuellen Realität.

²³Die x- und y-Achse spannen die Bodenebene auf.



(a) Modellierungswerkzeug



(b) 3D Visualisierung

Abbildung 3.7: d³FACT Bildschirmfotos

Bewertung des Entwicklungsstands von d³FACT

Die Modellierungsmethode von d³FACT ist grundsätzlich für diese Arbeit ausreichend, die nötigen Änderungen können relativ flexibel und leicht über eine Änderung der DTD integriert werden. Da die Zustandsänderungen in Ereignisroutinen frei gestaltet werden können und keine vordefinierten Komponentenklassen existieren, die den Zustand in bekanntem Maß verändern, ist eine automatisierte Modellanalyse sehr schwierig. Hier müssen die Zustandsänderungen klarer strukturiert werden. Die Erstellung einiger komplexer Modelle zum Testen aller Methoden der geregelten Vereinfachung ist mit dem bestehenden Modellierungswerkzeug hinreichend effizient möglich. Die Methoden der Modellmodifizierung zur Rückwärtssimulation sind ein guter Startpunkt für die Entwicklung einer automatisierten Vereinfachung durch Zusammenfassung. Die Visualisierung der Simulation in der VR ist ausreichend und muss nicht angepasst werden. Die Methode der Zustandsabbildung ist für eine automatische Abbildung nicht optimal und sollte komplett überarbeitet werden. Die Implementierung der dvdS in Bezug auf die Modellstruktur muss nur leicht angepasst werden, so dass Gruppen von Komponenten Gruppen von Komponenten anderer Komplexitätsstufen zugewiesen werden können. Methoden zur Partitionierung und Hierarchisierung, Vereinfachung und Regelung sind nicht vorhanden und müssen neu erstellt werden.

4 Zu leistende Arbeit

Wir können alles schaffen genau
wie die toll dressierten Affen
wir müssen nur wollen
wir müssen nur wollen

*(Müssen nur Wollen,
Wir sind Helden)*

In diesem Kapitel soll eine Aufstellung darüber erfolgen, welche Arbeiten im Verlauf dieser Dissertation nötig sind, um die in der Problemstellung definierten Ziele und Anforderungen zu erfüllen. Durch die Erarbeitung des Stands der Technik erschließt sich, dass zur Zielerreichung nicht alle nötigen Methoden neu entwickelt werden müssen, einige Anforderungen können durch bestehende Methoden erreicht werden, andere durch eine Anpassung bestehender Methoden.

Die Modellbeschreibung von d³FACT (Abschnitt 3.1.3) ist nicht ausreichend für die Durchführung einer dvdS mit einer vorgeschalteten geregelten Vereinfachung. Hierzu muss die Modellbeschreibung angepasst werden, indem einige Aspekte der Modellbeschreibung von Mueck einfließen und eine Menge standardisierter Komponentenklassen zur Automatisierbarkeit der Modellanalyse erstellt werden.

Die vorgestellte Methode der Entity Structure/Model Base Framework (Abschnitt 3.2.2) ist für diese Arbeit ausreichend, um eine Hierarchie von Teilmodellen unterschiedlicher Komplexität zu verwalten, und soll in die zu entwickelnde Modellbeschreibung einfließen. Zur Partitionierung soll ein Multilevel-Algorithmus (Abschnitt 3.2.1) verwendet werden, welcher sehr gute Ergebnisse auch auf sehr großen Graphen ermöglicht. Als Vergrößerungsmethode soll wie bei Karpyis und Kumar Heavy Clique Matching eingesetzt werden, weil es bessere Ergebnisse liefert als das Heavy Edge Matching und die Erfahrungsbasis bei Matchingalgorithmen insgesamt höher ist als bei den Alternativen, bspw. agglomeratives

Clustering. Aus Gründen der Einfachheit soll die Vergrößerungsphase so ausgedehnt werden, dass kein Partitionierungsschritt notwendig ist. Zur Verfeinerung soll das von Karpis und Kumar vorgeschlagene Greedy Refinement eingesetzt werden, da es bei kleinem Aufwand gute Ergebnisse verspricht. Damit diese Methoden zur Erfüllung der Anforderungen eingesetzt werden können, muss eine Zielfunktion für die Verfeinerung und für diese eine Kantengewichtungsfunktion entwickelt werden.

Zur Modellvereinfachung soll ein Verfahren konzipiert werden, welches eine Weiterentwicklung der Arbeiten aus Abschnitt 3.3.3 und eine Spezialisierung auf d³FACT darstellt. Dazu soll ein Regelsatz erstellt werden, der angibt, welche Komponenten in welchen Strukturen zu weniger Komponenten in einfacheren Strukturen zusammengefasst werden können. Wichtiger Aspekt dieses Regelsatzes muss die Abbildung von Parameterwerten des Ausgangsmodells auf Parameter der Vereinfachung sein, damit die Verhaltensabweichung möglichst gering bleibt. Zur Definition der nötigen Strukturen soll neben der Arbeit von Brooks und Tobias auch die Arbeit von Laroque (Abschnitt 3.6.3) benutzt werden. Zur Erzielung guter Vereinfachungen soll der Engpass in Strukturen nach Brooks und Tobias berechnet werden. Da die Komponentenklassen in dieser Arbeit von denen von Brooks und Tobias abweichen, muss die Engpassermittlung angepasst werden. Die in dieser Arbeit verwendeten deutlich komplexeren Komponentenklassen führen auch zu Erweiterungsbedarf bei der Zusammenfassung. Mit dem Regelsatz allein ist eine automatisierte Vereinfachung nicht möglich, deshalb muss eine Methode zur Modellanalyse entwickelt werden, mit der zusammenfassbare Strukturen in einem Modell gefunden werden können, und Methoden zur regelungsabhängigen Auswahl von Zusammenfassungstechniken aus dem Regelsatz sowie Anwendung dieser zur Synthese neuer Komponenten.

Um möglichst viele Vereinfachungspotentiale aufzudecken, sollen neben der Zusammenfassung auch noch die von Zeigler vorgestellten Methoden Weglassung von Komponenten und die Variablentransformation untersucht werden (vgl. Abschnitt 3.3.1). Dazu ist im Bereich Weglassung ein auf die Komponentenklassen und Strukturen bezogener Regelsatz zu erstellen, der festlegt, wann Weglassungen mit welchen Auswirkungen auf das Modellverhalten erfolgen können. Auch im Bereich Weglassung müssen, wie im Bereich der Zusammenfassung, Methoden zur Modellanalyse und -synthese entwickelt werden. Da die Verwendung von Simulationsmetamodellen nicht in Betracht kommt und in der Vereinfachung nach Brooks und Tobias auch die Vereinfachung durch Verwendung einer Modellhierarchie zur Beschränkung der Einflussgrößen implizit verwendet wird¹, sind alle

¹Bei allen Komponenten einer Zusammenfassung ausgenommen dem Engpass erfolgt eine Beschrän-

in der Taxonomie von Frantz (Abschnitt 3.3.1) identifizierten Vereinfachungsmöglichkeiten ausgeschöpft.

Ein Algorithmus zur Regelung der Vereinfachung in Bezug auf Komplexität und Verhaltensabweichung konnte nicht gefunden werden. Zwar existiert in der Arbeit von Johnson et al. eine Methode zur Regelung anhand der Verhaltensabweichung (Abschnitt 3.3.3), diese betrachtet nur eine Verhaltenskennzahl und führt zu einer sehr schlechten Laufzeit der Vereinfachung. Methoden zur Messung der Komplexität von Simulationsmodellen existieren (Abschnitt 3.4.1), sind aber weder für den Bereich Regelung der Vereinfachung noch für die Verwendung von Komponentenklassen entwickelt worden. Es soll eine Erweiterung des Maßes von Schruben und Yücesan erfolgen, mittels dessen die Komplexität der Komponenten durch Zyklen im Modellgraph gewichtet wird. Die Komplexität der Komponenten soll durch eine Anpassung des Maßes von Wallace berechnet werden. Zur Verhaltensmessung soll eine kleine Menge von Kennzahlen des Produktionscontrollings verwendet werden, die das Verhalten in Bezug auf die Ziele der Materialflusssimulation effektiv messen. Zur Bestimmung der Verhaltensabweichung soll ein Abstandmaß konzipiert werden, welches auf die Trajektorien der Kennzahlen anwendbar ist. Der Algorithmus der Regelung, der aus den Messergebnissen für Komplexität und Verhaltensabweichung Entscheidungen trifft und den Vereinfachungsalgorithmus steuert, muss gänzlich neu entwickelt werden.

Methoden zur Generierung von Zuständen zum Austausch von Teilmodellen während der Durchführung der dvdS existieren in einer Form, wie es die Nachsimulation von Mueck darstellt (Abschnitt 3.5.3). Eine Methode der Zustandsgenerierung, die wie die direkte Berechnung von Mueck funktioniert, muss von Grund auf neu entwickelt werden. Dazu müssen alle möglichen Zustände der Komponentenklassen ermittelt werden, eine Nachvollziehbarkeit der Vereinfachung hergestellt werden und ein Algorithmus erstellt werden, der die eigentliche Zustandsabbildung durchführt.

Sind alle Neu- und Weiterentwicklungen konzipiert, sollen sie implementiert und in das Simulationswerkzeug d³FACT integriert werden. Im Zuge der Implementierung sollen auch Anwendungen erstellt werden, die für eine (halb-) automatische Durchführung von empirischen Versuchen zur Validierung genutzt werden können.

kung der Einflussgrößen, da bspw. für alle durch Puffer substituierten Maschinen die Losgrößen und Rüstzeiten keine Beachtung mehr finden.

5 Konzeption

I'm gonna make him an offer
he can't refuse.

*(Don Vito Corleone,
The Godfather)*

In diesem Kapitel soll ein Konzept für ein automatisiertes Verfahren zur Modellvereinfachung vorgestellt werden, welches ein Ausgangsmodell hierarchisch partitioniert, die Komplexität der entstandenen Partitionen gezielt unter Validitätsgesichtspunkten reduziert, in die erzeugten Teilmodelle Zustandsabbildungsfunktionen integriert und somit eine Modellhierarchie im Sinne der dvdS erzeugt.

5.1 Übersicht

Abbildung 5.1 zeigt einen Entwurf des Gesamtsystems und wie sich die geregelte Vereinfachung in das System der detaillierungsvarianten diskreten Simulation (dvdS) einbettet.

Ausgehend vom Ausgangsmodell wird durch eine iterative Partitionierung des Ausgangsmodells eine Hierarchie von Partitionen erstellt. Aus jeder Partition entsteht durch eine Vereinfachung der umfassten Modellkomponenten ein Teilmodell geringerer Komplexität. Liegen für alle gebildeten Partitionen vereinfachte Teilmodelle vor, ist also ein vollständiges dvdS Modell vorhanden, kann die Simulation gestartet werden. Sollen während der Simulation Teilmodelle ausgetauscht werden, wie es die dvdS vorsieht, dann müssen für die neu zu aktivierenden Teilmodelle durch eine Zustandsberechnung gültige Zustände erzeugt werden. Dazu werden die für jedes Teilmodell erzeugten Zustandsabbildungen verwendet.

Abbildung 5.2 skizziert die Regelungsmethode der Vereinfachung. Zur Vereinfachung werden, sofern vorhanden, die schon erzeugten Teilmodelle höherer Komplexität $B_j(b_{i,h})$ ver-

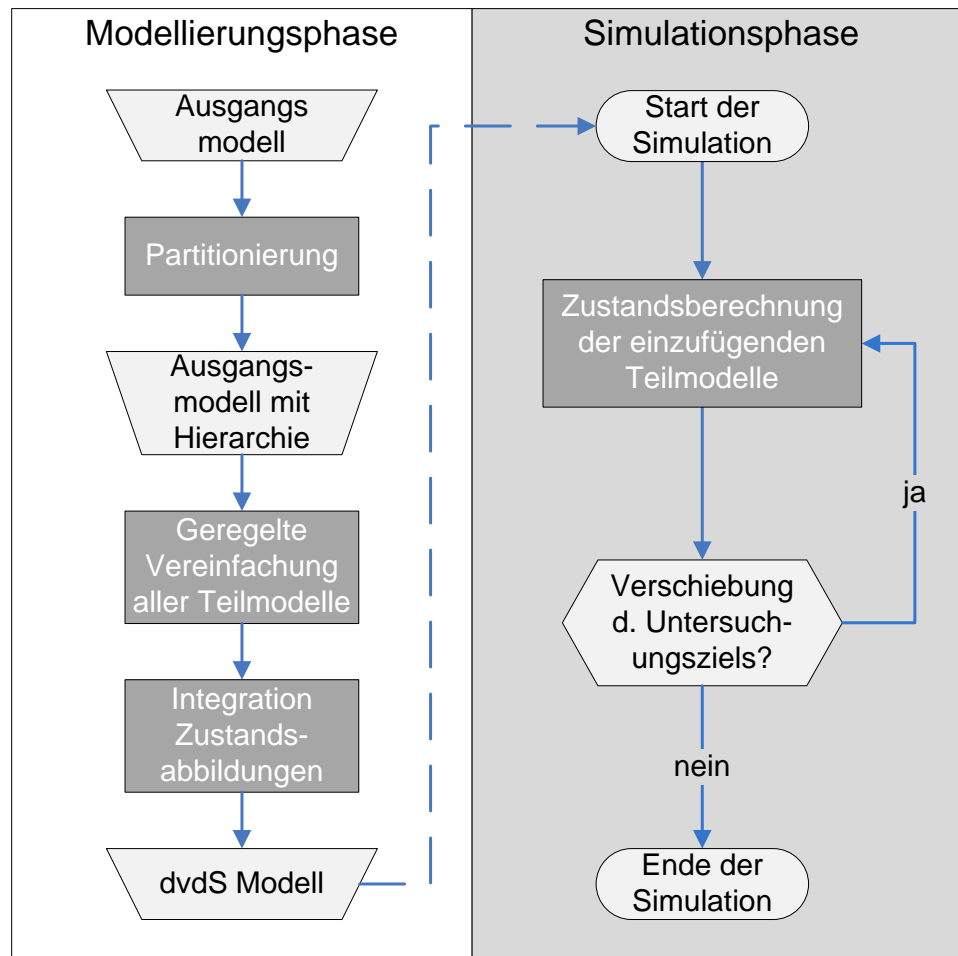


Abbildung 5.1: Verfahrensübersicht des Gesamtsystems

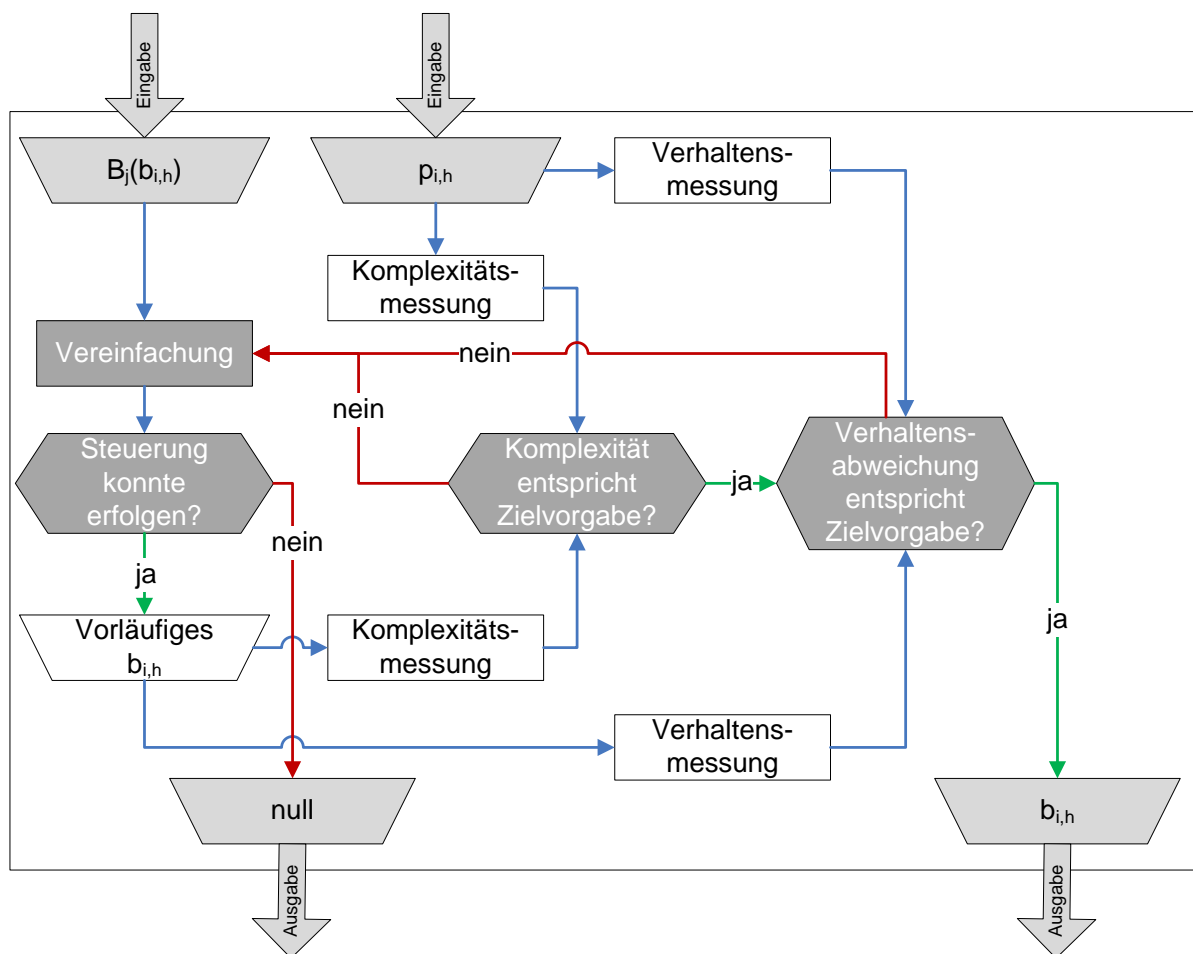


Abbildung 5.2: Regelung der Vereinfachung anhand Komplexität und Validität

wendet, die denselben Teil des Systems abbilden wie das zu erzeugende Teilmodell $b_{i,h}$. Die Maße zur Messung der Komplexität und Verhaltensabweichung verwenden zum Vergleich mit den ermittelten Werten für $b_{i,h}$ die Partition $p_{i,h}$ des Ausgangsmodells, die denselben Teil des Systems abbildet wie $b_{i,h}$. Durch die Verwendung von $B_j(b_{i,h})$ und $p_{i,h}$ wird zum einen die schon getätigte Vereinfachung der Teilmodelle in $B_j(b_{i,h})$ genutzt und der Aufwand für die Erstellung von $b_{i,h}$ verringert und zum anderen die Ausbreitung von Folgefehlern in den Messungen vermieden. Die Vereinfachung ist eingebunden in eine doppelte Rückkopplungsschleife aus Komplexitäts- und Verhaltensmessung. Werden die gestellten Anforderungen an ein zu erzeugendes Teilmodell von dem vorläufigen Teilmodell nicht erfüllt und konnte bei einer weiteren Durchführung der Vereinfachung keine Maßnahme getroffen werden, die die Eigenschaften des vorläufigen Teilmodells in Bezug zu den Zielvorgaben verbessert, dann kann kein Teilmodell $b_{i,h}$ erstellt werden. Können die Zielvorgaben erfüllt werden, kann das erzeugte Teilmodell in die Hierarchie eingefügt werden.

In den nächsten Abschnitten soll der Entwurf der nötigen Methoden für dieses Vorgehen detailliert beschrieben werden. Als Ausgangspunkt dazu muss als erstes eine Modellbeschreibungsmethode definiert werden.

5.2 Konzept einer Modellbeschreibungsmethode

Die Anforderungen an die Modellbeschreibung (Abschnitt 2.2.1) verlangen eine Methode, die automatisiert analysierbar ist, eine hohe Abbildungsflexibilität besitzt und sich möglichst wenig von dem bestehenden Konzept von d³FACT unterscheidet. Im nächsten Abschnitt wird dazu eine Menge von Komponentenklassen definiert, aus deren Instanzen das Ausgangsmodell gebildet werden muss. Auch alle generierten Teilmodelle bestehen aus Instanzen dieser Klassen. Danach wird dargestellt, wie ein für diese Arbeit verwendetes Modell beschrieben wird.

5.2.1 Festlegung der Menge von Modellkomponentenklassen

Soll ein Modell automatisiert analysiert und modifiziert werden, wie es durch Vereinfachung, Regelung und Zustandsgenerierung geschehen soll, so muss sein Aufbau standardisiert sein (vgl. Abschnitt 2.2.1). Eine Standardisierung soll hier, wie in den meisten

Simulationswerkzeugen üblich, durch die Definition einer Menge von parametrierbaren Komponentenklassen geschehen. Werden nur Instanzen dieser Klassen in einem Modell verwendet, so ist bekannt, welche Zustände sie einnehmen können und wie ihre Ereignisse den Zustand verändern. Bei der Vereinfachung ist es so möglich festzulegen, welche Gruppen von Komponenten durch kleinere Gruppen von anderen Komponenten substituiert werden können. Eine Zuordnung solcher Gruppen ist möglich, weil abgeschätzt werden kann, wie sich das Verhalten der Simulation durch die Substitution ändert. Bei der Zustandsgenerierung können der Zustand und die eingeplante Zustandsveränderung in der Zukunft einer Gruppe erfasst werden und auf eine andere Gruppe übertragen werden.

Bei der Festlegung der Menge an Modellkomponentenklassen soll darauf geachtet werden, dass die Menge groß genug ist, um eine große Anzahl praxisrelevanter Probleme mit diesen Komponenten modellieren zu können (Anforderung 2). Gleichzeitig soll die Menge auch möglichst klein sein, um die Existenz möglichst vieler Komponenten gleicher Klasse in einem Modell zu ermöglichen und so eine effektive Vereinfachung zu gewährleisten. Die Komponenten werden so gewählt, dass klassische Ziele der Materialflusssimulation für unterschiedliche Fertigungssysteme von Stückgütern untersucht werden können. Ein mit den Komponenten modelliertes Fertigungssystem soll folgende Eigenschaften haben können (vgl. [DW97]):

- Erzeugnisspektrum: Standarderzeugnisse mit und ohne Varianten
- Auslösungsart: nach Bestellung oder auf Lager
- Erzeugnisstruktur: ein- oder mehrteilig
- Dispositionsart: überwiegend oder rein programmorientiert
- Beschaffungsart: jeder Grad von Fremdbezug
- Fertigungsart: Kleinserie bis Massenfertigung
- Fertigungsablaufart: Werkstatt-, Gruppen oder Fließfertigung mit Abstraktion von Unstetigförderern
- Fertigungsstruktur: mittlere bis hohe Tiefe

Einschränkungen ggü. der vollständigen Darstellung von Dangelmaier [DW97] entstehen daraus, dass nicht jeder Artikel - jede Marke - absolut einzigartig¹ sein kann, weil ansons-

¹Es muss eine Schnittmenge von Eigenschaften geben, die auf jeder Marke vorhanden sind. Diese Eigenschaften können aber unterschiedliche Ausprägungen haben.

ten die Steuerungsregeln des Materialflusses nicht standardisierbar sind. Die Fertigungsmittel müssen feste Positionen haben, weil sonst keine Partitionierung möglich ist, und eine gewisse Strukturtiefe muss vorhanden sein, damit sich eine Vereinfachung überhaupt lohnt.

Die zu modellierende Menge an Fertigungssystemen deckt sich mit der, die auch mit den vordefinierten Komponentenklassen bekannter Simulationswerkzeuge wie eM-Plant² oder Witness³ modelliert werden kann. So wurde die Menge an Komponentenklassen an die der genannten Simulationswerkzeuge angelehnt: Quellen zum Erzeugen von Marken, Senken zum Vernichten, Maschinen und Puffer als klassische Elemente der Warteschlangentheorie und Verteiler und Zusammenführer zur Steuerung des Materialflusses. Für eine Modellierung realistischer Fertigungssysteme sind Förderbänder zum zeitverzehrenden Fördern vorgesehen und Sequenzer, um die Güter des Materialflusses anhand ihrer Bezeichnung in ihrer Reihenfolge zu sortieren.

Zur Modellierung vieler Prozesse im Materialfluss sind Zufallsverteilungen nötig, deshalb haben viele der definierten Komponentenklassen zufallsverteilte Parameter. Häufig benutzte Verteilungen sind u.a. die Normal-, Exponential- und Dreiecksverteilung. Diese Verteilungen werden auch für die Parameter der Komponenten in der Vereinfachung verwendet, weil beim Entwurf der Vereinfachungsmethoden berücksichtigt wird, keine Methoden einzusetzen, wie bspw. Faltung⁴, die spezielle Verteilungen erzeugen.

Alle erstellten Komponentenklassen enthalten nur Variablen und in Ereignisroutinen definierte Logik, die für die materialflussbezogene Funktion der Komponenten nötig ist. Die in Abschnitt 2.2 definierte Anforderung der Schlankheit der Modellierung wird demnach erfüllt.

Quelle

In dieser Modellkomponente werden Marken erzeugt. Alle Marken erhalten eine Beschreibung und einen Zeitstempel des Erzeugungstermins. Als Statistikkenzahl wird in jeder Quelle der Durchsatz gezählt.

²Neuerdings unter dem Namen Plant Simulation, Produkthomepage: www.emplant.de

³Produkthomepage: www.lanner.com

⁴Die Faltung erzeugt eine Funktion, die die „Überlappung“ von zwei anderen Funktionen darstellt, bspw. angewendet für die Addition von Zufallsverteilungen. Vgl. Abschnitt 5.4.1

Quelle Intervall; QI

In dieser Ausführung der Quelle werden Marken mit einer konstanten Beschreibung erzeugt. Zwei Erzeugungen trennt ein Zeitintervall.

Parametervariablen: Markenbeschreibung, Zeitintervall zufallsverteilt oder als deterministischer Wert

Quelle Verteilung; QV

In dieser Ausführung existiert eine Liste von unterschiedlichen Beschreibungen und dazugehörige Erzeugungswahrscheinlichkeiten. Diese Quelle erzeugt demnach Marken mit unterschiedlichen Beschreibungen. Die Markenmenge je Beschreibung über die Zeit wird durch die Erzeugungswahrscheinlichkeit definiert. Zwei Erzeugungen trennt ein Zeitintervall.

Parametervariablen: Liste mit Markenbeschreibung und Erzeugungswahrscheinlichkeiten, Zeitintervall zufallsverteilt oder als deterministischer Wert

Quelle Liste; QL

Hier existiert eine Liste von Erzeugungszeitpunkten und Markenbeschreibungen, mit der erreicht werden kann, dass spezielle Marken, d.h. Marken mit sehr unterschiedlichen Beschreibungen, definiert und zu vorher festgelegten Zeitpunkten erzeugt werden können. Bei dieser Quelle ist es zudem möglich, den Marken einen Zeitstempel aufzuprägen, der angibt, wann die Marke in einer Senke ankommen soll. Dieser Zeitstempel kann als geplanter Liefertermin interpretiert werden. So kann eine Priorisierung in Anlehnung an die Schlupfzeit in Puffern erfolgen.

Parametervariablen: Tabelle mit Erzeugungszeitpunkten, Markenbeschreibungen und Liefertermin

Quelle Logisch; Q

Diese Quellenklasse wird nur in Kombination mit Demontagemaschinen eingesetzt, der einzigen Komponente im System, aus der mehr Marken austreten als eintreten. Diese Differenz wird durch eine logische Quelle erzeugt.

Parametervariablen: keine

Senke

In dieser Modellkomponente werden Marken vernichtet.

Senke Statistik; SS

Hier werden die Durchlaufzeit und der Durchsatz von Marken als statistische Kennzahlen gebildet.

Parametervariablen: keine

Senke Logisch; S

Nicht alle in Senken vernichtete Marken stellen Faktoren dar, die zur Bestimmung der Kennzahlen Durchlaufzeit und Durchsatz benutzt werden sollen. In diesem Fall werden logische Senken verwendet. Sie werden bspw. an Montagestationen angeschlossen, um die Anbauteile logisch zu vernichten.

Parametervariablen: keine

Maschine

Diese Modellkomponente verzögert Marken in ihrem Fluss durch das Modell und kann die Beschreibung von Marken ändern, um „bearbeitete“ Marken von „unbearbeiteten“ unterscheidbar zu machen. Die Verzögerung stellt die Bearbeitungszeit der Maschine dar und sorgt dafür, dass die Marke(n), um eine Zeit nach dem Beginn der Bearbeitung in die Maschine verzögert, wieder aus ihr austreten. Allen Maschinentypen ist gemein, dass die Bearbeitungszeit zufallsverteilt oder deterministisch sein kann und dass sie ausfallen können, d.h. für eine Zeitspanne nicht zu Verfügung stehen. Ein Rüstvorgang, d.h. eine verlängerte Verzögerung, wird eingeleitet, wenn eine zur Bearbeitung eintretende Marke nicht dieselbe Beschreibung aufweist wie die letzte bearbeitete Marke. In jeder Maschinenklasse werden als Statistik Kennzahlen der Durchsatz, die Auslastung, die kumulierte Arbeits-, Rüst- und Störzeit erfasst.

Maschine Normal; M

Diese Maschinenklasse verzögert eine Marke oder ein Los von Marken für eine festgelegte Zeit. Die Bearbeitungszeit beginnt, wenn genug Marken für ein Los in der Maschine angekommen sind. Die Losgröße wird für eine Maschineninstanz festgelegt und bleibt während der Simulation konstant.

Parametervariablen: Bearbeitungszeit zufallsverteilt oder als deterministischer Wert, Losgröße, Rüstzeit zufallsverteilt oder als deterministischer Wert, MTTF und MTTR zufallsverteilt oder als deterministischer Wert, Markenbeschreibung

Maschine Montage; MM

Eine Montagemaschine hat mehrere Eingänge, die auch Marken zu Losen sammeln können. Sind in allen Eingängen den Losgrößen entsprechend viele Marken eingegangen, beginnt die Bearbeitungszeit, nach der die Marken die Maschine verlassen. In dieser Maschine ist ein „Fertigteileingang“ definiert. Marken, die über diesen Eingang in die Maschine eintreten, verlassen die Maschine auch wieder normal. Alle anderen Marken werden als Anbauteile betrachtet und in einer angeschlossenen logischen Senke vernichtet.

Parametervariablen: Bearbeitungszeit zufallsverteilt oder als deterministischer Wert, Losgröße für jeden Eingang, Rüstzeit zufallsverteilt oder als deterministischer Wert, MTTF und MTTR zufallsverteilt oder als deterministischer Wert, Markenbeschreibung

Maschine Demontage; MD

Eine Demontagemaschine besitzt mehrere Ausgänge, denen ähnlich den Eingängen der Montagestation Losgrößen zugeordnet sind. Kommen der Losgröße des Eingangs entsprechend viele Marken an, so treten nach der Bearbeitungszeit den Losgrößen der Ausgänge entsprechend viele Marken aus der Demontagestation aus. Die mengenmäßige Differenz an Marken zwischen Eingang und Ausgang wird durch eine angeschlossene logische Quelle ausgeglichen. Bei dieser Maschine ist ein „Fertigteilausgang“ definiert, bei dem die Marken die Komponente verlassen, die über den Eingang eingetreten sind.

Parametervariablen: Bearbeitungszeit zufallsverteilt oder als deterministischer Wert, Losgröße für den Eingang und die Ausgänge, Rüstzeit zufallsverteilt oder als deterministischer Wert, MTTF und MTTR zufallsverteilt oder als deterministischer Wert, Markenbeschreibung

Puffer mit Steuerstrategie; PF, PL oder PP

Puffer sind zeitlose Komponenten mit einer festgelegten Puffergröße, die angibt, wie viele Marken aufgenommen werden können. Zur Entnahme können mehrere Strategien eingesetzt werden. Bei der FiFo Methode (PF) kommt es zu keiner Umsortierung und die Marke, die sich am längsten im Puffer befindet, tritt als nächstes aus. Bei der LiFo Methode (PL) tritt eine Umsortierung auf, da die Marke, die die kürzeste Zeit im Puffer ist, als nächstes austritt. Anhand des Zeitstempels auf Marken, der den Liefertermin widerspiegelt, kann eine Umsortierung stattfinden, wenn die Prioritätsmethode als Steuerung des Puffers (PP) gewählt wird. Hier tritt die Marke als nächstes aus dem Puffer aus, die die kleinste dieser Lieferzeiten aufgeprägt hat. Es ergibt sich eine Priorisierung, die sich an die Schlupfzeit anlehnt. Als Statistikkennzahl werden in jedem Puffer der Durchsatz, der mittlere Bestand und die Wartezeit bestimmt.

Parametervariablen: Puffergröße

Sequenz; SQ

Der Sequenzer ist eine zeitlose Komponente ähnlich einem Puffer, die versucht, Sequenzen von Marken gleicher Beschreibung zu bilden. Sind der Sequenzlänge entsprechend viele Marken einer Beschreibung in den Sequenzer eingegangen, dann werden diese an den Nachfolger abgegeben. Der Einsatz eines Sequenzers ist möglich, um die Rüstzeiten von nachgelagerten Maschinen zu verringern oder um sortenreine Lose für nachgelagerte Maschinen mit einer Losgröße größer eins zu bilden. Als Statistikkennzahl werden der Durchsatz, der mittlere Bestand und die Wartezeit bestimmt.

Parametervariablen: Puffergröße, Sequenzlänge

Förderband; F

Ein Förderband ist ein spezieller Puffer, der nach der FiFo Methode arbeitet und zusätzlich noch zeitbehaftet ist. Marken können diese Modellkomponente erst verlassen, nachdem

eine Verzögerungszeit abgelaufen ist, die ihrer Förderzeit entspricht. Als Statistikkennzahl werden der Durchsatz, die Auslastung, der mittlere Bestand und die Wartezeit bestimmt.

Parametervariablen: Förderzeit als deterministischer Wert, Puffergröße

Verteiler; VV, VR oder VB

Ein Verteiler hat einen Eingang und mehrere Ausgänge, er ist zeitlos und seine Puffergröße ist immer eins. Anhand von einer aus drei möglichen Methoden wird entschieden, über welchen Ausgang eine Marke diese Komponente verlässt. Für die Verteilungs-Methode (VV) wird jedem Ausgang eine Wahrscheinlichkeit zugewiesen und mit Hilfe eines Zufallsgenerators wird festgelegt, welcher Ausgang gewählt wird. Wird die Reihum-Methode (VR) gewählt, wird versucht, über alle Ausgänge ungefähr gleich viele Marken zu verschicken. Für beide Verteilungsmethoden gilt, dass, wenn der gewünschte Ausgang belegt ist, ein anderer Ausgang gewählt wird. Sind alle Ausgänge belegt, so wird über den Ausgang geschickt, der als erstes frei wird. Durch dieses Verhalten wird gewährleistet, dass es zu keinen Verklemmungen kommen kann. In der Beschreibungs-Methode (VB) werden jedem Ausgang eine oder mehrere Markenbeschreibungen zugewiesen. Marken verlassen den Verteiler dann über den Ausgang, dem ihre Beschreibung zugewiesen wurde. So können Marken anhand ihrer Beschreibung getrennt werden.

Parametervariablen: Liste mit Wahrscheinlichkeiten für die Ausgänge, Liste mit Beschreibungen für die Ausgänge

Zusammenführer, Z

Diese Modellkomponente hat mehrere Eingänge und einen Ausgang, sie ist zeitlos und ihre Puffergröße ist immer eins. Alle eintreffenden Marken treten über den einen Ausgang wieder aus.

Parametervariablen: keine

5.2.2 Modellbeschreibung

Ein dvdS-Modell ist eine Menge B von Teilmodellen. Jedes Teilmodell $b \in B$ ist ein Knoten in einem Hierarchiebaum mit der Menge an Hierarchieebenen H . Jeder Knoten der Hierarchie ist entweder in der Menge \hat{B} der aktiven oder in der Menge \check{B} der inaktiven Teilmodelle. Jedes Teilmodell hat festgelegte Schnittstellen, die den Markenein- und Markenausgang des Teilmodells darstellen.

Jedes Modell besteht aus einer Menge Komponenten V . Jede Komponente $v \in V$ ist eine Instanz einer Klasse d aus der Menge der Komponentenklassen D . Jede Komponente hat eine Position $o \in \mathbb{R}^2$ und durch die Klasse geerbte dreidimensionale Form f , Ereignismenge E_d , Zustandsvariablen I_d und Channel Ch_d . Eine Adjazenzmatrix $\Psi = (ch) \times (ch) \rightarrow \{0, 1\}$ stellt die Verknüpfungen zwischen den Komponenten über ihre Channels dar.

Zur Laufzeit der dvdS existiert, initiiert durch den Ausgangszustand $s_0(\hat{B})$, eine Liste Q , für einen Zeitpunkt t , eingeplanter Ereignisse (e, t, b, v) . Diese Ereignisse werden zu ihren eingeplanten Zeitpunkten ausgeführt und verändern den Zustand $s(\hat{B}) = (Q, I(v \in \hat{B}), M)$, bestehend aus der Liste der eingeplanten Ereignisse, den Werten der Zustandsvariablen und der Markenbelegung M .

Mit der so gewählten Modellbeschreibung sind die Änderungen zu d³FACT minimal, wie in Anforderung 4 gefordert. Änderungen sind die Restriktion auf den \mathbb{R}^2 und somit auf eine Modellfläche. Dies ist einer einfacheren Partitionierung geschuldet, entspricht aber auch den Gegebenheiten in der Realität, da Fertigungssysteme im Allgemeinen in einer Ebene aufgebaut werden. Die zweite Änderung betrifft die Hierarchie als die Organisationsstruktur der Teilmodelle und die auf Teilmodelle bezogenen Bestandteile des Simulationszustands.

5.3 Konzept einer Partitionierungsmethode

In diesem Abschnitt soll eine Zielfunktion für den Verfeinerungsschritt in der Multilevel-Partitionierungsmethode und eine Kantengewichtungsfunktion entwickelt werden. Der ausgeglichene Umfang der Partitionen wird über eine Balance-Restriktion eingestellt, die genau dann erfüllt ist, wenn alle Partitionen gleich groß sind. Um die beiden anderen Ziele der Partitionierung nicht zu stark zu behindern, ist die Balance-Restriktion eine weiche Restriktion. Die beiden anderen Ziele sind der logische Zusammenhang, der durch eine

Minimierung der Schnittgröße erreicht wird, und der geometrische Zusammenhang, erzielt durch die Kantengewichtungsfunktion. Die Qualität einer n -Partition $p = \{V_1, \dots, V_n\}$ soll anhand des Ratio-Cuts (bspw. Chan et al. [CSZ93])

$$rc = \frac{cut(p)}{\prod_{i=1}^n |V_i|} \quad (5.1)$$

gemessen werden. Dieser bewertet sowohl die Balance als auch die Schnittgröße und soll deshalb invertiert ($\frac{1}{rc}$) als Qualitätsmaß der Partitionierung verwendet werden.

Um diese Qualitätsfunktion während des Partitionierens zu maximieren, bedarf es einer Zielfunktion. Die Vergrößerungsphase arbeitet mit dem HCM indirekt auf die Erfüllung der Anforderungen hin, indem sie den Graphen so vergrößert, dass anschließend nur noch sehr leichte Kanten geschnitten werden können. In der Verfeinerungsphase, bei dem verwendeten vereinfachten KL-Algorithmus, liegt das größte Verbesserungspotential und deshalb soll für diesen Schritt eine Zielfunktion entwickelt werden.

Der KL-Algorithmus verwendet standardmäßig folgende Zielfunktion:

$$\begin{aligned} gain(v_i) &> 0; \\ |V_1| - |V_2| &> \omega(v_i) \cdot BF, \end{aligned} \quad (5.2)$$

wobei BF der Balancing-Faktor ist, der normalerweise den Wert 1 besitzt. Wenn diese Zielfunktion erfüllt ist, dann ist das Verschieben eines Knoten v_i von Partition V_1 nach V_2 erlaubt.

Als mögliche Verbesserung dieser einfachen Zielfunktion wird eine weitere Zielfunktion definiert, die im Rahmen der Validierung mit der normalen KL-Zielfunktion verglichen werden soll. Diese Zielfunktion erlaubt Tauschoperationen, wenn die Qualität der Balance verbessert wird. Die Qualität der Balance ist dabei definiert als:

$$\sigma(p) = \sqrt{\frac{1}{|V|} \cdot \sum_{i=1}^p \left(|V_i| - \frac{|V|}{n} \right)^2}, \quad q(p) = cut(p) \cdot \sigma(p)$$

wobei $\sigma(p)$ die Standardabweichung der Knotenzahl einer Partition vom angestrebten Mittel $\frac{|V|}{n}$ ist und λ ein vom Nutzer festgelegter Schwellenwert. Somit ergibt sich die

Zielfunktion als:

$$q(p_{pre}) - q(p_{post}) > \frac{|V|}{\lambda} \quad (5.3)$$

Für die Bildung von Partitionen mit geometrischem Zusammenhang wird eine Kantengewichtung im Verhältnis zum Quadrat ihrer Länge bestimmt, so dass längere Kanten eher geschnitten werden als kurze.

$$\omega((u, v)) = \left(\frac{1}{\sqrt{(o_x(v) - o_x(w))^2 + (o_y(v) - o_y(w))^2}} \right)^2 \quad (5.4)$$

5.4 Konzeption von Methoden zur automatisierten Modellvereinfachung

Die Vereinfachung soll sich an den Arbeiten von Zeigler, Brooks und Tobias, Johnson et al., Völker und Hung und Leachman (siehe Abschnitt 3.3.1 und 3.3.3) orientieren, weil die verwendeten Methoden zur Erfüllung wesentlicher an die Vereinfachung gestellter Anforderungen geeignet scheinen. Hung und Leachman zeigen, dass eine starke Vereinfachung bei gleichzeitig kleiner Verhaltensabweichung möglich ist, Johnson et al. zeigen, dass eine Regelung möglich ist, Völker zeigt, dass eine fast stufenlose Vereinfachung möglich ist, und Brooks und Tobias zeigen, dass ein rein modellanalytisches Vorgehen möglich ist.

Durch die Zusammenfassung von Komponenten, auch aufzufassen als die Substitution von großen Komponentengruppen durch kleinere, soll im Sinne der Reduzierung der ausmodellierten Anlagen die Zahl der Ereignisse pro Simulationszeit bei unverändertem Produktionsprogramm reduziert werden (vgl. [HL99]). Neben der Zusammenfassung, die den stärksten Hebel zur Vereinfachung aufweist, sollen auch die von Zeigler genannten Methoden der Vereinfachung durch Variablentransformation und Weglassung auf ihre Anwendbarkeit untersucht werden. Stellt sich später die Variablentransformation als recht wirkungslos heraus, so kann eine durch Zusammenfassung gebildete Vereinfachung durch Weglassung bereinigt und weiter vereinfacht werden.

Nachfolgend wird für die Methoden Zusammenfassung und Weglassung ein Regelsatz definiert, der genau vorschreibt, wie Komponenten zusammengefasst bzw. weggelassen werden können. Zudem wird für beide Methoden ein Algorithmus erstellt, der diese Regeln anwendet, um ein Modell automatisch zu vereinfachen.

5.4.1 Zusammenfassung von Modellkomponenten

Bei der Zusammenfassung sollen die Engstellen im Modell berücksichtigt werden, weil diese wesentlichen Einfluss auf die Verhaltensabweichung haben (vgl. Johnson et al. [JFM05]). Zur Identifizierung dieser wird die Engstellen-Analyse von Brooks und Tobias [BT00] auf die hier verwendete Modellbeschreibungsmethode angepasst und zur Verbesserung der Genauigkeit erweitert. Die neu definierte relative effektive Kapazität $C_{eff,rel}$ aller Komponenten in einem Teilmodell wird mit Hilfe der Formeln 5.5 berechnet.

Die Zusammenfassung betrachtet nicht ein komplettes Teilmodell, sondern Strukturen (vgl. Grundstrukturen bei Laroque [Lar07]), die Teile von Teilmodellen sind. In einem Teilmodell lassen sich also globale Engstellen definieren und lokale (je Struktur). Die relative effektive Kapazität ist für Komponenten der Maschinen- (V_M) und Förderbandklasse (V_F) definiert. Alle anderen Komponenten können somit keinen Engpass darstellen, weil sie nicht zeitbehaftet sind oder Quellen sind, welche als Systemgrenzen nicht als Engpässe betrachtet werden sollen.

$$\begin{aligned}
 V_{bn} &= \{v \in (V_M \cup V_F) \mid C_{eff,rel}(v) = \max_{w \in V} (C_{eff,rel}(w))\} \\
 0 \leq C_{eff,rel}(v) &= \frac{\min_{w \in V} (C_{eff,cyc}(w))}{C_{eff,cyc}(v)} \leq 1 \\
 C_{eff,cyc}(v) &= \frac{C_{eff}(v)}{\frac{Z_v(v)}{Z_{blc}}} \\
 C_{eff}(v) &= \frac{LS(v_M)}{t_U(v_M) + t_R(v_M)} \cdot \frac{MTTF(v_M)}{MTTR(v_M) + MTTF(v_M)} \\
 C_{eff}(v) &= \frac{BS(v_F)}{t_U(v_F)}
 \end{aligned} \tag{5.5}$$

Als Erweiterung zu Brooks und Tobias wurde durch die Definition von $C_{eff,rel}$ eine Normierung auf den Wertebereich $[0, 1]$ vorgenommen und diese sind somit einfacher automatisch zu vergleichen. Durch diese Normierung haben die globalen Engstellen immer einen Wert von $C_{eff,rel} = 1$. Eine lokale Engstelle ist dadurch ausgezeichnet, dass sie den größten Wert von $C_{eff,rel}$ in der betrachteten Komponentenmenge besitzt. Eine weitere Erweiterung ist die Integration des Zyklen-Quotienten, der solche Komponenten stärker gewichtet, die eine hohe Bedeutung im Materialfluss besitzen (siehe Abschnitt 5.5.2). Diese Erweiterung soll die automatisierte Auswahl der Engstellen in einem Modell unterstützen, da ansonsten Komponenten als Engstellen identifiziert würden, die keine sind. Dies trifft insbesondere für Komponenten in parallelen Linien auf, die zwar von ihrer Pa-

rametrierung her Engstellen sind, aber dadurch keine Engstellen sind, da sie nur ein Teil des Materialflusses durchfließt.

Wie bspw. bei Rose [Ros00] soll die Engstelle in der durch Zusammenfassung neu generierten Struktur weitgehend unverändert übernommen werden. Fast alle anderen Komponenten des komplexen Modells gehen in neu erzeugten Komponenten, meist Förderbändern, mit approximiertem kumuliertem Verhalten auf. Damit ein stochastisches Verhalten des Modells erhalten bleibt, soll auch eine Maschine⁵ je Struktur erhalten bleiben, auch wenn ein Förderband die Engstelle darstellt. Die spezielle Berücksichtigung des Engpasses hat zudem den Vorteil, dass keine Zufallsverteilungen addiert werden müssen. Dieses Faltung genannte Problem ist nicht allgemein lösbar. Für die Faltung von Zufallsverteilungen gibt es nur in einigen Sonderfällen Lösungen, bei denen das Ergebnis ebenfalls eine „normale“ Zufallsverteilung ist, meist entstehen neue Verteilungen oder es muss eine Zusammenstellung von mehreren Verteilungen erfolgen⁶. Die Normalverteilung stellt einen dieser Sonderfälle dar⁷. Neben der Normalverteilung ist die Exponentialverteilung eine häufig verwendete Verteilungsfunktion. Bei einer Zusammenfassung müssten also mit großer Wahrscheinlichkeit Normalverteilungen mit Exponentialverteilungen gefaltet werden. Dieses Problem ist nicht im Rahmen dieser Arbeit lösbar, noch werden dadurch Verteilungen entstehen, aus denen mit normalem Aufwand eine Zufallszahl bestimmt werden kann, weil es zusammengesetzte Verteilungen sind.

Um eine möglichst geringe Verhaltensabweichung zwischen Ausgangsmodell und Vereinfachung zu erreichen, soll das Störverhalten, die Verfügbarkeit (beschrieben durch MTTF und MTTR), der in die Vereinfachung übernommenen Engstelle angepasst werden. Wird eine Linie mit mehreren Maschinen durch eine Linie mit nur einer Maschine substituiert, dann muss die Verfügbarkeit dieser geringer sein als der Durchschnitt der Maschinen in der Ausgangslinie, weil die Ausgangslinie blockiert ist, wenn nur eine der Maschinen in ihr ausfällt. Wird eine parallele Struktur durch eine Linie mit nur einer Maschine substituiert, so muss die Verfügbarkeit dieser höher sein als der Durchschnitt der Maschinen in der parallelen Struktur, weil die parallele Struktur erst blockiert, wenn alle ihre Linien blockiert sind. Die Störparameter der übernommenen Engstellen werden mit folgenden

⁵Wie in Abschnitt 5.2.1 definiert, haben nur Maschinen und Quellen zufallsverteilte Parameter und auch Quellen bleiben im vereinfachten Modell erhalten.

⁶Als ein Beispiel für eine solche Zusammenfassung soll Cobb et al. [RCRS07] angeführt sein.

⁷ $N(\mu_1 + \mu_2, \sigma_1 + \sigma_2) = N_1(\mu_1, \sigma_1) + N_2(\mu_2, \sigma_2)$

Formeln aus der Zuverlässigkeitsanalyse (bspw. [Bir91]) berechnet:

$$\lambda^{lin}(V) = \sum_V \lambda(v) \quad (5.6)$$

$$A_{eff}^{lin}(V) = 1 - \left(\sum_V \frac{\lambda(v)}{\mu(v)} \right)$$

$$\lambda^{para}(V) = \frac{\prod_V \lambda(v) \cdot \sum_V \mu(v)}{\prod_V \mu(v)} \quad (5.7)$$

$$A_{eff}^{para}(V) = 1 - \frac{\prod_V \lambda(v)}{\prod_V \mu(v)}$$

$$\mu(V) = \frac{\lambda(V)}{1 - A_{eff}(V)} \quad (5.8)$$

Mit $\mu = \frac{1}{MTTR}$ und $\lambda = \frac{1}{MTTF}$ ergeben sich die Störparameter $MTTF_{eff}^{lin}$ (Formel 5.6) für lineare und $MTTF_{eff}^{para}$ (Formel 5.7) für parallele Strukturen. Der Wert für MTTR (Formel 5.8) wird mit Hilfe der effektiven Verfügbarkeit berechnet, die auch für lineare und parallele Strukturen unterschiedlich ist.

Die Vereinfachung soll nahezu stufenlos erfolgen können, dies soll erreicht werden, indem zu vereinfachende Teilmodelle in mehrere zusammenfassbare Strukturen unterteilt werden. Die Vereinfachung der Teilmodelle erfolgt dann durch eine iterative Zusammenfassung aller in ihr befindlichen Strukturen. Nach jeder Iteration kann überprüft werden, ob der letzte Schritt ein den Anforderungen entsprechendes Teilmodell generiert hat oder ob weitere Iterationsschritte nötig sind. Durch die Anzahl Strukturen in den Teilmodellen ergeben sich diskrete Schritte in der Vereinfachung, so dass die Vereinfachung nur fast stufenlos ist. Die Anzahl Strukturen in den Teilmodellen kann angepasst werden, indem in großen Strukturen (bspw. sehr lange Linien) mehrere zu erhaltene Engstellen definiert werden und Teilstrukturen um diese Engstellen herum einzeln zusammengefasst werden.

Zur automatisierten Vereinfachung durch Zusammenfassung müssen zwei Techniken konzipiert werden: zum einen die Erkennung von Strukturen (Modellanalyse) und zum anderen die Zusammenfassung an sich (Modellsynthese). Die Erkennung von Strukturen soll in Anlehnung an einen Algorithmus erfolgen, der von Laroque [Lar07] im Rahmen der Entwicklung einer automatischen Erstellung von Rückwärtssimulationen erarbeitet wurde. Die grundlegenden Strukturen sind linear verbundene Komponenten und die parallele Anordnungen linearer Strukturen.

Diese beiden elementaren Strukturen wurden gewählt, weil sie häufig anzutreffen sind und

weitere Strukturen bei Laroque aus diesen zusammengesetzt sind. Ihre wichtigste Eigenschaft für die Vereinfachung ist, dass sie abgeschlossen sind, so dass alle Marken, die an dem einzigen Eingang eintreten, aus dem einzigen Ausgang austreten. Diese Strukturen können durch eine oder eine Linie von Komponenten substituiert werden, ohne dass die Konsistenz des Modells zerstört wird. Die Strukturen Kreuzung, Stern, Verzweigung, Zusammenführung und Rückkopplung haben entweder mehrere Ein- und Ausgänge oder es besteht im Fall der Rückkopplung die Möglichkeit, dass Marken gar nicht austreten. Diese nicht abgeschlossenen Strukturen können nicht durch eine oder eine Linie von Komponenten substituiert werden und werden deshalb bei der Vereinfachung nicht berücksichtigt.

Sind Strukturen gefunden worden, so werden diese zusammengefasst, wodurch sich der Modellgraph ändert und u.U. neue Strukturen gefunden werden können (vgl. [Lar07]); ein iteratives Vorgehen ist somit möglich. Welche Strukturen zusammengefasst werden können und welche Komponenten daraus entstehen, soll im nächsten Abschnitt entwickelt werden.

Regelsatz der Zusammenfassung linearer Strukturen

Einführend sollen Linien untersucht werden, die homogen aus Maschinen, Puffern oder Förderbändern bestehen. Danach werden gemischte Linien und die Problematik die Sequenzer in Linien erzeugen untersucht. Montage- und Demontagemaschinen, die Beginn bzw. Ende einer Linie bilden können, werden anschließend untersucht. Abgeschlossen wird dieser Abschnitt mit der Einführung der „Minimallinie“, die die maximale Zusammenfassung einer linearen Struktur darstellt.

Linie von Maschinen

Gegeben ist eine Linie im Teilmodell mit, vom Eingang zum Ausgang hin, durchnummerierten Maschinen M_1 bis M_n . Die Engstelle M_z wurde nach Formel 5.5 identifiziert und es gilt $1 \leq z \leq n$. Bis auf die mittlere Dauer bis zum Ausfall MTTF, die mittlere Reparaturdauer MTTR, die mit Formeln 5.6 bis 5.8 berechnet werden, und die Beschreibung ausgehender Marken bleibt die Maschine M_z unverändert. Zur Bewahrung der Validität des Modells (bspw. nachfolgende Verteiler nach dem Beschreibungsprinzip) muss die Engpassmaschine in der Vereinfachung die Marken so beschreiben, wie die letzte Maschine der Linie im Teilmodell. Die Bearbeitungs- und Rüstzeiten der Maschinen vor und hinter der Engstelle werden durch neue Förderbänder kompensiert, die in der Vereinfachung

vor und hinter dem übernommenen Engpass platziert werden. Die Losgrößenpuffer der Maschinen werden durch die Puffergrößen der Förderbänder kompensiert (s. Tabelle 5.1). Ist die Engpassmaschine die erste oder letzte Maschine der Struktur, entfällt entweder F1 oder F2.

Die Kompensation der Rüstzeit ist ein optionaler Term, der von der Regelung der Vereinfachung bei Bedarf weggelassen werden kann. Ist das Produktionsprogramm recht homogen, d.h. wird im Ausgangsmodell sehr selten gerüstet, dann bietet sich die Nichtberücksichtigung an. Da in der Zusammenfassung kein Zugriff auf das Produktionsprogramm besteht, kann diese Entscheidung nur von der Regelung durch simulative Verhaltensmessung getroffen werden.

Die gewählte Zusammenfassungsregel minimiert den durch Vereinfachung induzierten Fehler in Bezug auf die drei Kennzahlen Durchlaufzeit, Umlaufbestand und Durchsatz⁸. Die Summe aller Zeiten, die ein Auftrag im Ausgangsmodell bearbeitet oder gefördert wird, ist erhalten geblieben, so dass die Durchlaufzeit erhalten bleibt. Ebenso bleibt die Gesamtlagerkapazität erhalten, so dass der Umlaufbestand gleiche Werte annehmen kann. Die konkrete Ausprägung der Abweichungen von Durchlaufzeit und Umlaufbestand sowie dem Durchsatz hängt allerdings in entscheidendem Maße vom Produktionsprogramm und der Auslastung der Struktur ab. Durch eine hohe Auslastung entstehen Wartezeiten von Aufträgen auf Maschinen, die betragsmäßig nicht bei der Vereinfachung berücksichtigt werden können, weil sie nicht explizit im Modell notiert sind. Da der Engpass der Struktur erhalten bleibt, sollten auch die Wartezeiten relativ gleich bleiben, eine Abweichung im Verhalten ist aber nicht zu vermeiden.

Linie von Förderbändern

Gegeben ist eine Linie im Ausgangsmodell mit vom Eingang zum Ausgang durchnummerierten Förderbändern F1 bis Fn. Eine Zusammenfassung kann dann wie in Tabelle 5.2 dargestellt erfolgen. Die Zusammenfassung einer Linie von Förderbändern kann auf zwei Weisen erfolgen, abhängig davon, ob ein Förderband der Struktur eine andere effektive Kapazität $C_{eff,rel}$ aufweist als die übrigen. In diesem Fall ist ein Engpass Fz in der Struktur vorhanden und muss entsprechend berücksichtigt werden. In der Realität ist allerdings davon auszugehen, dass linear verbundene Förderbänder aufeinander abgestimmt sind und somit kein Engpass existiert. Existiert kein Engpass in der Struktur, dann kann

⁸Detaillierte Ausführungen hierzu in Abschnitt 5.5.3

Tabelle 5.1: Zusammenfassung von zwei oder mehr Maschinen Normal in einer Linie

M1 - ... - Mn	Komponentenparameter
	$t_U(\cdot)$ $LS(\cdot)$ $t_R(\cdot)$ $MTTF(\cdot)$ $MTTR(\cdot)$ $mb(\cdot)$
F1 - Mz - F2	Zusammenfassungskomponenten und -parameter
Mz	$t_U = t_U(M_{bn})$
Mz	$LS = LS(M_{bn})$
Mz	$t_R = t_R(M_{bn})$
Mz	$MTTF = MTTF_{eff}$
Mz	$MTTR = MTTR_{eff}$
Mz	$mb = mb(Mn)$
F1	$t_U = \sum_{i=1..z-1} t_U(Mi) \left[+ \sum_{i=1..z-1} t_R(Mi) \right]$
F1	$BS = \sum_{i=1..z-1} LS(Mi)$
F2	$t_U = \sum_{i=z+1..n} t_U(Mi) \left[+ \sum_{i=z+1..n} t_R(Mi) \right]$
F2	$BS = \sum_{i=z+1..n} LS(Mi)$

Tabelle 5.2: Zusammenfassung von zwei oder mehr Förderbändern in einer Linie

F1 - ... - Fn	Komponentenparameter
	$BS(\cdot)$ $t_U(\cdot)$
F	Zusammenfassungskomponenten und -parameter
	$BS = \sum_{i=1..n} BS(i)$ $t_U = \sum_{i=1..n} t_U(Fi)$
F1 - Fz - F2	Zusammenfassungskomponenten und -parameter
Fz	$BS = BS(F_{bn})$
Fz	$t_U = t_U(F_{bn})$
F1	$BS = \sum_{i=1..z-1} BS(i)$
F1	$t_U = \sum_{i=1..z-1} t_U(Fi)$
F2	$BS = \sum_{i=z+1..n} BS(i)$
F2	$t_U = \sum_{i=z+1..n} t_U(Fi)$

Tabelle 5.3: Zusammenfassung von Puffern in einer Linie

Komponenten	Zusammenfassung
PF - PF	PF
PL - PL	PL
PP - PP	PP
PL - PF	PL
PP - PF	PP
PL - PP	PL - PP
PP - PL	PP - PL

die Zusammenfassung zu einem einzigen Förderband erfolgen. Ansonsten besteht die vereinfachte Struktur aus drei Förderbändern und die Vereinfachung erfolgt analog zu der linearen Struktur von Maschinen.

Linie von Puffern

Linien von Puffern werden im Allgemeinen nicht im Modell vorhanden sein. Eine Betrachtung ist aber trotzdem sinnvoll, da mehrere, auch unterschiedliche Puffer in heterogenen Linien auftreten und im Zuge der Vereinfachung direkt miteinander verbunden werden können. Linien von Puffern können durch paarweise Zusammenfassung vereinfacht werden, somit sind mit den festgelegten Komponentenklassen die in Tabelle 5.3 dargestellten Paarungen möglich. Der einzige Parameter von Puffern ist die Puffergröße, diese wird bei einer Zusammenfassung kumuliert. Ein FiFo-Puffer nimmt keine Umsortierung in der Reihenfolge der eintreffenden Aufträge vor, diese erfolgt aber in einem LiFo-Puffer bestimmt und in einem Prioritätspuffer unter Umständen⁹. Um die Auswirkungen auf das Verhalten möglichst klein zu halten, muss diese Umsortierung erhalten bleiben, so dass die Zusammenfassung eines FiFo-Puffers mit einem Puffer einer anderen Klasse immer in einem Puffer der anderen Klasse resultiert. Eine Zusammenfassung von LiFo- und Prioritätspuffern ist nicht möglich.

Linie mit Puffern, Förderbändern und Maschinen

Gegeben ist eine Linie im Ausgangsmodell mit vom Eingang zum Ausgang durchnummerierten Komponenten 1 bis n der Klassen Maschine Normal, Puffer FiFo und Förderband. Die Engstelle M_z wurde nach Formel 5.5 identifiziert und es gilt $1 \leq z \leq n$. Eine Zusammenfassung kann dann wie in Tabelle 5.4 dargestellt erfolgen und ist die Kombination

⁹Unter der realistischen Bedingung, dass die Belegung größer als 1 werden kann.

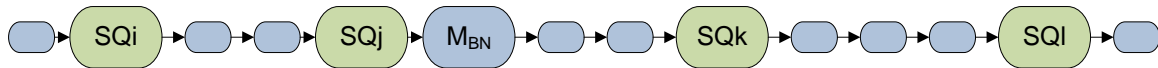


Abbildung 5.3: Mehrere Sequenzer in zusammenzufassender Linie

der oben vorgestellten Zusammenfassungen. Sollte in der untersuchten Struktur keine Maschine den Engpass darstellen, sondern ein Förderband F_y , dann muss das Engpass-Förderband erhalten bleiben (s. unterer Abschnitt von Tabelle 5.4). Es wird dann analog zu Tabelle 5.2 parametrisiert und je nach Position in der Ausgangsstruktur vor oder hinter M_z platziert. Die Maschine M_z muss erhalten bleiben, auch wenn sie nicht die Engstelle in der Struktur darstellt, damit das stochastische Verhalten erhalten bleibt, insbesondere in Hinblick auf das Störverhalten. Für den Spezialfall $t_U(F1) = 0$ oder $t_U(F2) = 0$ wird abweichend ein FiFo-Puffer verwendet anstatt eines Förderbandes ohne Förderzeit. Dies gilt auch für alle anderen hier definierten Zusammenfassungsregeln, um die Komplexität zu minimieren (vgl. hierzu die C_v in Tabelle 5.8).

Gemischte Linie mit Sequenzern

Sequenzer verlangen nach einer besonderen Behandlung, weil sie zum einen den Materialfluss umsortieren und zum anderen Wartezeiten induzieren, die vom Produktionsprogramm verursacht werden. Umfasst das Produktionsprogramm viele unterschiedliche Markensorten, dann ist die Wartezeit in einem Sequenzer, bis genug Marken für ein Los eingetroffen sind, im Allgemeinen größer als wenn das Produktionsprogramm nur wenige unterschiedliche Markensorten umfasst. Existieren Markensorten, die nur sehr selten auftreten, dann wird dieser Effekt noch vergrößert. Sequenzer sind für die Steuerung des Materialflusses vor einer Maschine oder Maschinenlinie konzipiert, müssen also immer im Kontext zu einer solchen betrachtet werden. Existieren in einer Linie Sequenzer vor der Engstelle (vgl. Abbildung 5.3), dann muss der letzte Sequenzer vor der Engstelle erhalten bleiben. Existieren Sequenzer hinter der Engstelle, dann muss der letzte Sequenzer der Linie erhalten bleiben, um Lose für Maschinen zu bilden, die außerhalb der untersuchten Struktur liegen. Wenn ein Sequenzer substituiert wird, dann wird er behandelt wie ein Puffer, indem seine Puffergröße auf das entsprechende Förderband addiert wird.

Linie mit Montagemaschine

Eine Montagemaschine führt mehrere Linien zu einer zusammen und kann somit nicht sub-

Tabelle 5.4: Zusammenfassung von gemischten Linien

	Komponentenparameter
Mi	$t_U(\cdot)$
Mi	$LS(\cdot)$
Mi	$t_R(\cdot)$
Mi	$MTTF(\cdot)$
Mi	$MTTR(\cdot)$
Mi	$mb(\cdot)$
PFi	$BS(\cdot)$
Fi	$t_U(\cdot)$
Fi	$BS(\cdot)$
F1 - Mz - F2	Zusammenfassungskomponenten und -parameter
Mz	$t_U = t_U(M_{bn})$
Mz	$LS = LS(M_{bn})$
Mz	$t_R = t_R(M_{bn})$
Mz	$MTTF = MTTF_{eff}$
Mz	$MTTR = MTTR_{eff}$
Mz	$mb = mb(Mi_{max})$
F1	$t_U = \sum_{i=1..z-1} t_U(Mi) + \sum_{i=1..z-1} t_U(Fi) [+ \sum_{i=1..z-1} t_R(Mi)]$
F1	$BS = \sum_{i=1..z-1} LS(Mi) + \sum_{i=1..z-1} BS(Fi) + \sum_{i=1..z-1} BS(PFi)$
F2	$t_U = \sum_{i=z+1..n} t_U(Mi) + \sum_{i=z+1..n} t_U(Fi) [+ \sum_{i=z+1..n} t_R(Mi)]$
F2	$BS = \sum_{i=z+1..n} LS(Mi) + \sum_{i=z+1..n} BS(Fi) + \sum_{i=z+1..n} BS(PFi)$
F1 - Fy - Mz - F2	Zusammenfassungskomponenten und -parameter
Mz	wie gehabt
Fy	$BS = BS(F_{bn})$
Fy	$t_U = t_U(F_{bn})$
F1	$t_U = \sum_{i=1..z-1} t_U(Mi) + \sum_{i=1..z-1 \setminus i=y} t_U(Fi) [+ \sum_{i=1..z-1} t_R(Mi)]$
F1	$BS = \sum_{i=1..z-1} LS(Mi) + \sum_{i=1..z-1 \setminus i=y} BS(Fi) + \sum_{i=1..z-1} BS(PFi)$
F2	$t_U = \sum_{i=z+1..n} t_U(Mi) + \sum_{i=z+1..n \setminus i=y} t_U(Fi) [+ \sum_{i=z+1..n} t_R(Mi)]$
F2	$BS = \sum_{i=z+1..n} LS(Mi) + \sum_{i=z+1..n \setminus i=y} BS(Fi) + \sum_{i=z+1..n} BS(PFi)$

Tabelle 5.5: Zusammenfassung einer Montagemaschine mit einer gemischten Linie

MM1 - ...	Komponentenparameter
MM1	t_U
MM1	$LS_{in1}, \dots, LS_{inm}$
MM1	t_R
MM1	$MTTF$
MM1	$MTTR$
MM1	mb
Mi	$t_U(\cdot)$
Mi	$LS(\cdot)$
Mi	$t_R(\cdot)$
Mi	$MTTF(\cdot)$
Mi	$MTTR(\cdot)$
Mi	$mb(\cdot)$
PFi	$BS(\cdot)$
Fi	$t_U(\cdot)$
Fi	$BS(\cdot)$
MMz - F1	Zusammenfassungskomponenten und -parameter
MMz	$t_U = t_U(M_{bn})$
MMz	$LS_{in1}, \dots, LS_{inm}$
MMz	$t_R = t_R(M_{bn})$
MMz	$MTTF = MTTF_{eff}$
MMz	$MTTR = MTTR_{eff}$
MMz	$mb = mb(Mi_{max})$
F1	$t_U = \sum_{\forall i \setminus z} t_U(Mi) + \sum_{\forall i \setminus z} t_U(Fi) \left[+ \sum_{\forall i \setminus z} t_R(Mi) \right]$
F1	$BS = \sum_{\forall i \setminus z} LS(Mi) + \sum_{\forall i \setminus z} BS(Fi) + \sum_{\forall i \setminus z} BS(PFi)$

stituiert werden. Sie kann aber mit Komponenten in der Linie hinter ihr zusammengefasst werden (s. Tabelle 5.5). Da in diesem Fall nur die Engstelle übernommen werden kann, wenn die Montagemaschine diese ist, kann die Montagemaschine auch die Eigenschaften der Engstelle erhalten¹⁰.

Linie mit Demontagemaschine

Eine Demontagemaschine teilt eine Linie in mehrere auf und kann somit nicht substituiert werden. Sie kann aber mit Komponenten in der Linie vor ihr zusammengefasst werden (s. Tabelle 5.6). Da in diesem Fall nur die Engstelle übernommen werden kann, wenn die

¹⁰Auch in diesem Fall ist es möglich, dass ein Förderband den Engpass darstellt. Ist dies der Fall, so muss das Förderband erhalten bleiben (siehe Tabelle 5.4). Der Übersichtlichkeit halber wird hier aber auf eine getrennte Darstellung dieser Situation verzichtet.

Tabelle 5.6: Zusammenfassung einer Demontagemaschine mit einer gemischten Linie

... - MDn	Komponentenparameter
MDn	t_U
MDn	LS
MDn	$LS_{out1}, \dots, LS_{outm}$
MDn	$t_{R1}, t_{R2}, \dots, t_{Rn}$
MDn	$MTTF$
MDn	$MTTR$
MDn	$mb_{MD1}, \dots, mb_{MDm}$
Mi	$t_U(\cdot)$
Mi	$LS(\cdot)$
Mi	$t_R(\cdot)$
Mi	$MTTF(\cdot)$
Mi	$MTTR(\cdot)$
Mi	$mb(\cdot)$
PFi	$BS(\cdot)$
Fi	$t_U(\cdot)$
Fi	$BS(\cdot)$
F1 - MDz	Zusammenfassungskomponenten und -parameter
MDz	$t_U = t_U(M_{bn})$
MDz	$LS = LS_{M_{bn}}$
MDz	$LS_{out1}, \dots, LS_{outm}$
MDz	$t_R = t_R(M_{bn})$
MDz	$MTTF = MTTF_{eff}$
MDz	$MTTR = MTTR_{eff}$
MDz	$mb_{MD1}, \dots, mb_{MDm}$
F1	$t_U = \sum_{\forall i \setminus z} t_U(Mi) + \sum_{\forall i \setminus z} t_U(Fi) \left[+ \sum_{\forall i \setminus z} t_R(Mi) \right]$
F1	$BS = \sum_{\forall i \setminus z} LS(Mi) + \sum_{\forall i \setminus z} BS(Fi) + \sum_{\forall i \setminus z} BS(PFi)$

Demontagemaschine diese ist, kann die Demontagemaschine auch die Eigenschaften der Engstelle erhalten.

„Minimallinie“

Enthalten Linien keine Montage- oder Demontagemaschinen, dann kann jede Linie bis zur „Minimallinie“ zusammengefasst werden. Eine weitere Zusammenfassung ist nicht möglich. Die „Minimallinie“ ergibt sich wie folgt¹¹:

¹¹Im konkreten Fall einer Vereinfachung enthält die „Minimallinie“ natürlich nur die Komponenten, die sich aus dem Ausgangsmodell ergeben. Ist bspw. in der Struktur des Ausgangsmodells kein Sequenzer vorhanden, dann enthält auch die Vereinfachung keinen.

PP1 - PL1 - SQ1 - F1 - Fy - Mz - F2 - PP2 - PL2 - SQ2

Zusätzlich zu den oben beschriebenen Regeln wird bei den zu erhaltenden Puffern beachtet, ob sie vor oder hinter dem Engpass liegen, es kann also vorkommen, dass mehrere Puffer derselben Art erhalten bleiben müssen. Das mögliche Engpassförderband tritt hingegen nur einmal in der Minimallinie auf, kann aber vor oder hinter der Engpassmaschine liegen. An der Minimallinie wird deutlich, dass die maximal mögliche Vereinfachung sehr stark vom Ausgangsmodell abhängt. Befinden sich in vielen Strukturen zu erhaltene Komponenten, so enthält die Vereinfachung deutlich mehr Komponenten.

Regelsatz der Zusammenfassung paralleler Strukturen

Parallele Strukturen sind eine Menge „Minimallinien“ zwischen einem Verteiler und einer Zusammenführung. Alle Marken, die in den Verteiler eintreten, treten nach einer gewissen Zeit aus der Zusammenführung wieder aus. Linien, die von einem Verteiler Reihum oder Verteilung aufgespannt werden, können zu einer Linie zusammengefasst werden, wenn sie den gleichen strukturellen Aufbau haben, was im Folgenden als homogene Linien bezeichnet werden soll. Ist dies nicht für alle von einem Verteiler aufgespannten Linien der Fall, dann kann eine Teilzusammenfassung der Untermenge homogener Linien erfolgen. Linien, die von einem Verteiler mit Beschreibungsmethode aufgespannt werden, können nicht zusammengefasst werden, da ein solcher Verteiler nur eingesetzt werden wird, wenn die Linien nicht homogen sind.

Gegeben seien mehrere homogene „Minimallinien“; ein Verteiler mit der Nummer 1 und ein Zusammenführer mit der Nummer n und alle Komponenten der „Minimallinien“ sind nummeriert $1 < i < n$ und in jeder Linie gelte, dass die Nummer des Nachfolgers einer Komponente i die Nummer $j = i + 1$ besitzt. Dann kann eine parallele Struktur mit $r = 1..m$ Linien wie in Tabelle 5.7 dargestellt zusammengefasst werden. Um das Verhalten der resultierenden Linie ähnlich dem der parallelen Linien zu gestalten, gibt es zwei Möglichkeiten, die Parameter von Maschinen zu wählen. Zum einen könnte die Losgröße der resultierenden Maschine auf die Summe aller Losgrößen der Maschinen gesetzt werden. In diesem Fall würden für die Bearbeitungs- und Rüstzeit die Mittelwerte über alle Maschinen gesetzt werden. Zum anderen könnte für die Losgröße der Mittelwert genommen werden und für die beiden Zeiten der Mittelwert mal $\frac{1}{m}$, wie in Tabelle 5.7 dargestellt. Im zweiten Fall wird die Gesamtkapazität der Struktur verändert, weil die Losgrößen-Puffer

Tabelle 5.7: Zusammenfassung homogener Linien

V1 < Minimallinien > Zn	Komponentenparameter
V1	$BS = 1$
Zn	$BS = 1$
PP1r, PL1r	$BS(\cdot)$
PP2r, PL2r	$BS(\cdot)$
SQ1r, SQ2r	$BS(\cdot)$
F1r, F2r	$BS(\cdot)$
F1r, F2r	$t_U(\cdot)$
Mr	$t_U(\cdot)$
Mr	$LS(\cdot)$
Mr	$t_R(\cdot)$
Mr	$MTTF(\cdot)$
Mr	$MTTR(\cdot)$
Mr	$mb(\cdot)$
Minimallinie	Zusammenfassungskomponenten und -parameter
PP1	$BS = \sum_{\forall PP1r} BS(\cdot)$
PL1	$BS = \sum_{\forall PL1r} BS(\cdot)$
PP2	$BS = \sum_{\forall PP2r} BS(\cdot)$
PL2	$BS = \sum_{\forall PL2r} BS(\cdot)$
SQ1	$BS = \sum_{\forall SQ1r} BS(\cdot)$
SQ2	$BS = \sum_{\forall SQ2r} BS(\cdot)$
F1	$BS = \sum_{\forall F1r} BS(\cdot)$
F1	$t_U = \overline{t_U(F1r)}$
F2	$BS = \sum_{\forall F2r} BS(\cdot)$
F2	$t_U = \overline{t_U(F2r)}$
M	$t_U = \frac{1}{m} \overline{t_U(Mr)}$
M	$LS = \overline{LS(Mr)}$
M	$t_R = \frac{1}{m} \overline{t_R(Mr)}$
M	$MTTF = MTTF_{eff}$
M	$MTTR = MTTR_{eff}$
M	$mb = mb(Mi_{max})$

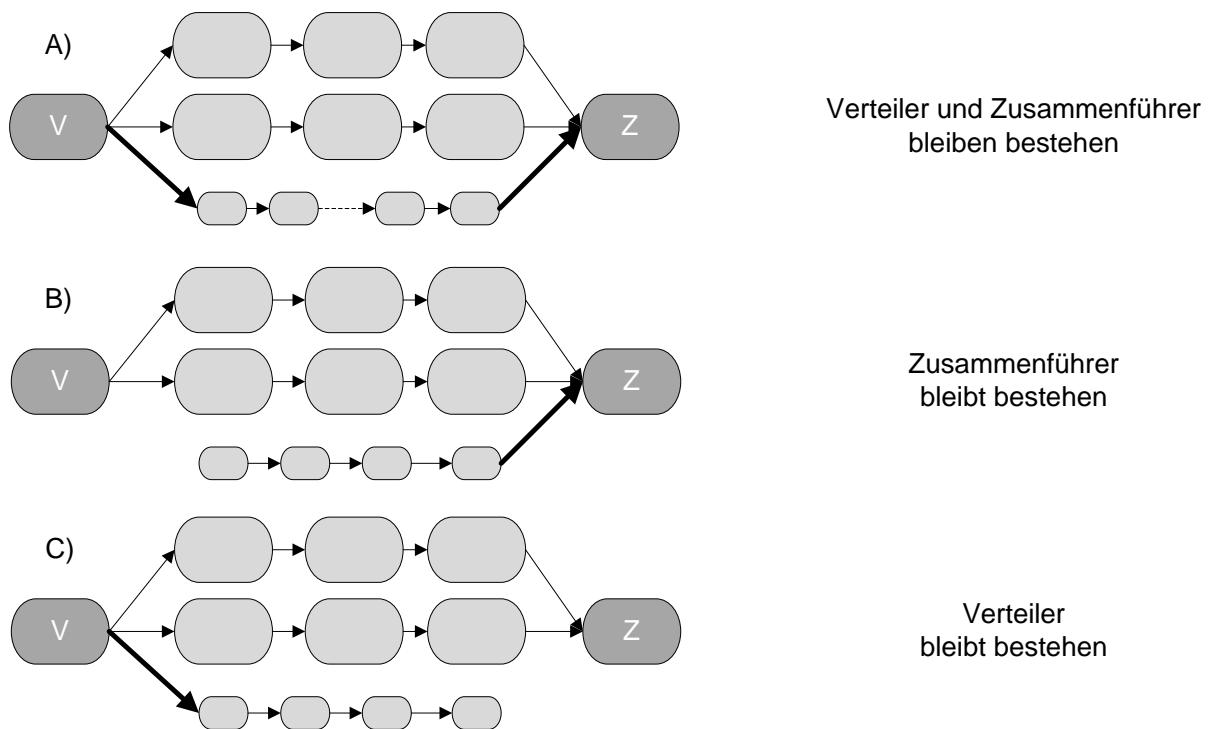


Abbildung 5.4: Mehrere Sequenzer in zusammenzufassender Linie

von $m - 1$ Maschinen verloren gehen. Dieser Ansatz wurde trotzdem gewählt, weil das Risiko besteht, dass Instabilitäten wegen ungeeigneter Puffergrößen vor und nachgelagerter Puffer entstehen. Die Puffer im Ausgangsmodell sind auf eine Losgröße von $\overline{LS}(Mr)$ ausgelegt, bei einer Losgröße von $\sum_{r=1..m} LS(Mr)$ im vereinfachten Modell könnte es zu Blockieren und Hungern von Maschinen kommen, wo es dieses im Ausgangsmodell nicht gibt (vgl. Abschnitt 5.4.3).

Zusammenfassungen von Untermengen einer parallelen Linie, bzw. entarteten parallelen Linien, können in den in Abbildung 5.4 dargestellten Fällen vorgenommen werden, wenn als Verteiler die Methoden Reihum oder Verteilung genutzt werden. Die Zusammenfassung der homogenen Untermenge erfolgt wie in Tabelle 5.7 dargestellt. In den Fällen, in denen der Verteiler erhalten bleibt, müssen die Verteilungsregeln angepasst werden. Die durch Zusammenfassung entstandene Linie muss im Verteiler eine Ausgangswahrscheinlichkeit haben, die der Wahrscheinlichkeit entspricht, mit der irgendeine Linie der homogenen Untermenge gewählt würde. Damit dieses geschehen kann, muss ein Verteiler Reihum durch einen Verteiler Verteilung ersetzt werden.

Modellanalyse - Finden von Strukturen

Durch die Modellanalyse soll in zu vereinfachenden Teilmodellen eine Struktur gefunden werden, die mittels Zusammenfassung vereinfacht werden kann. Die Modellanalyse wird von der Regelung der Vereinfachung angesteuert.

Algorithmus 1 (Modellanalyse Zusammenfassung).

```

1: function ANALYSEZUSAMMENFASSUNG( $b$ ,  $BN$ )
2:   Bilde Lineare Strukturen aus unmarkierten Komponenten und Parallele Strukturen
   aus markierten Komponenten, unter Beachtung der Engstellenerhaltung
3:   Liste  $L_1 = \{\text{Strukturs} | v \in s \wedge v \in BN\}$ 
4:   Liste  $L_2 = \{\text{Strukturs} | s \notin L_1\}$ 
5:   Sortiere beide Listen nach der Komplexität der Strukturen
6:   for all Strukturen in  $L_2$  do
7:     if Struktur zusammenfassbar then
8:       return Struktur
9:     end if
10:  end for
11:  for all Strukturen in  $L_1$  do
12:    if Struktur zusammenfassbar then
13:      return Struktur
14:    end if
15:  end for
16:  return null
17: end function

```

Die Modellanalyse benötigt ein Teilmodell b und die Menge der zu erhaltenen Engstellen BN als Eingabe. Im Algorithmus der Modellsynthese werden Komponenten markiert, so dass im Laufe der Vereinfachung die Anzahl markierter Komponenten steigt. Aus unmarkierten Komponenten werden lineare Strukturen gebildet werden und aus markierten parallele Strukturen. Lineare Strukturen werden so gebildet, dass jede Struktur maximal eine Komponente aus der Menge der zu erhaltenen Engpässe beinhaltet, die in der Regelung definiert wird. In parallelen Linien kann in jeder zusammenzufassenden Linie eine zu erhaltene Engstelle vorhanden sein. Je mehr Engpässe zu erhalten sind, umso größer ist die Zahl der Strukturen. Die Erhöhung der Anzahl Strukturen erfolgt durch die Zerteilung

linearer Strukturen. Es gibt aber auch Strukturen, in denen kein Engpass liegt. Weil diese weniger Einfluss auf die Verhaltensabweichung haben werden, werden diese bevorzugt zur Zusammenfassung ausgewählt. Durch die Sortierung der Strukturen nach ihrer Komplexität wird versucht, die Zielkomplexität mit möglichst wenigen Zusammenfassungsschritten zu erreichen, was sich positiv auf die Zustandsabbildung auswirkt, weil mehr Komponenten unverändert im zu aktivierenden und zu deaktivierenden Modell vorhanden sind (siehe Abschnitt 5.6).

Modellsynthese - Zusammenfassen von Strukturen

In der Modellsynthese wird die Struktur, die die Modellanalyse ausgewählt hat, durch Zusammenfassung vereinfacht und eine neu generierte Struktur zurückgeliefert. Dazu soll folgender einfacher Algorithmus verwendet werden:

Algorithmus 2 (Modellsynthese Zusammenfassung).

- 1: **procedure** SYNTHESZUSAMMENFASSUNG(b , *Struktur*, $\{C_{eff,rel}(v) | v \in b\}$)
- 2: *Wähle Zusammenfassungsmethode aus dem Regelsatz aus*
- 3: *Erzeuge Ausgabestruktur*
- 4: *Berechne Parametrierung der Ausgabestruktur mit $\{C_{eff,rel}(v) | v \in \text{Struktur}\}$*
- 5: *Markiere Komponenten der Ausgabestruktur als untersucht*
- 6: *Tausche in b die Struktur mit der Ausgabestruktur aus*
- 7: **end procedure**

5.4.2 Transformation von Variablen

Die Vereinfachung durch Variablentransformation verändert den Typ einer Variablen. Anstatt einer Zufallsverteilung wird für die Variable der Erwartungswert der Verteilung als deterministischer Wert festgelegt. Werden nur deterministische Variablen eingesetzt, so verliert die Simulation ihr charakteristisches zufallsabhängiges Verhalten. Die Vereinfachung durch Variablentransformation hat somit Einfluss auf das Verhalten der Vereinfachung, welcher umso stärker ist, je größer die Varianz der substituierten Verteilung ist. Von der Transformation von Variablen als Mittel zur Senkung der Komplexität kann allerdings kein großer Hebel ausgehen. Wie in Abbildung 5.5¹² zu sehen, ist der Auf-

¹²Messungen erfolgen auf einem Intel P4 3GHz und 3GB RAM unter Fedora 7 mit dem Simulationswerkzeug d³FACT.

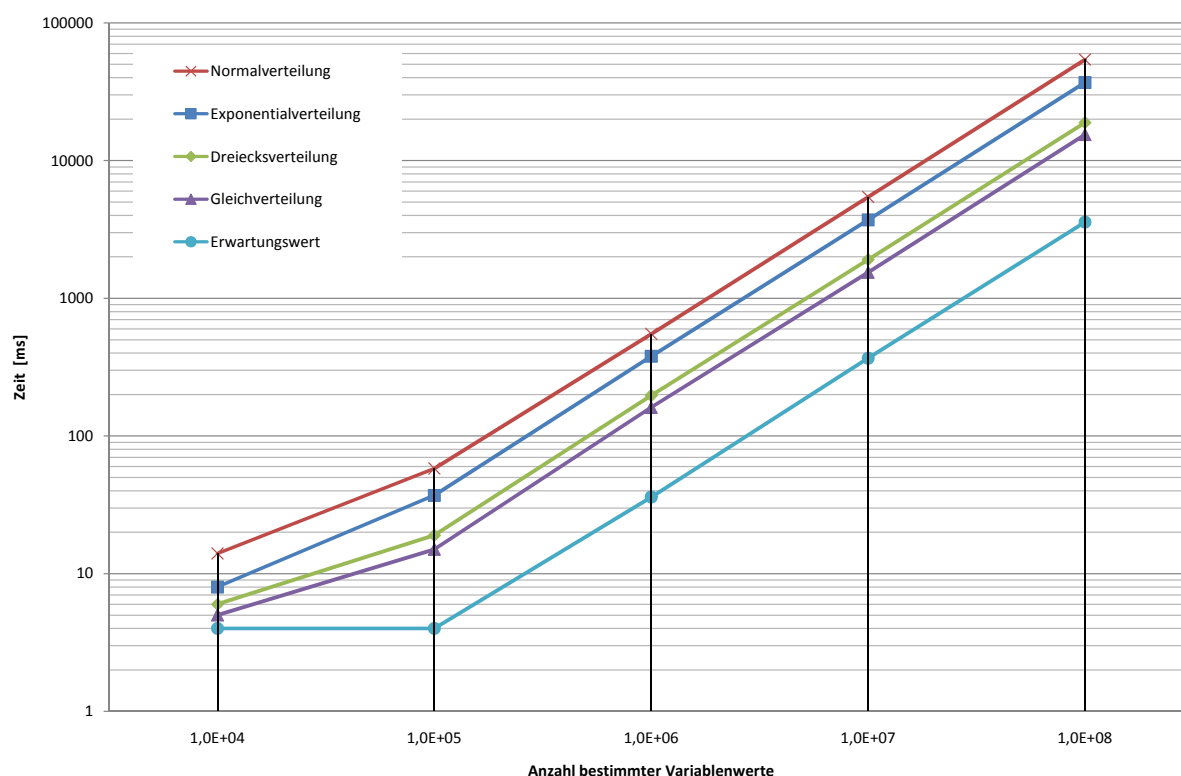


Abbildung 5.5: Zeitmessung für Bestimmung von Zufallsvariablen

wand zur Bestimmung des Erwartungswertes zwar um bis zu Faktor 15 niedriger¹³ als die Bestimmung der Zufallszahl. Die absoluten Werte sind allerdings klein, so dauert die Berechnung von 10^6 Zufallszahlen nur maximal 500 ms. Selbst in sehr großen Modellen, bei denen sehr viele Zufallszahlen benötigt werden, wird die mögliche Einsparung unterhalb des einstelligen Prozentbereichs liegen¹⁴. Die Transformation von Variablen soll somit nicht zur Vereinfachung genutzt werden.

5.4.3 Weglassung von Komponenten

Durch Weglassungen kann das durch Zusammenfassung vereinfachte Modell weiter vereinfacht werden, indem die Zahl der im Modell existierenden Komponenten weiter reduziert wird. Im Folgenden soll, wie bei der Zusammenfassung, ein Regelsatz erstellt werden, der die Auswirkungen einer Weglassung bewertet und festlegt, welche Komponenten bei

¹³In Bezug zur Normalverteilung als aufwendigste. Aufgrund von Messungenauigkeiten konnten für die Bestimmung von weniger als 10^5 keine signifikanten Werte gemessen werden.

¹⁴Anhand von Erfahrungswissen begründete Schätzung ohne Angabe eines Beweises.

welchen Zielvorgaben in Bezug auf erlaubte Verhaltensabweichung weggelassen werden können.

Bei jeder Weglassungsentscheidung muss geprüft werden, wie groß die zu erwartende Verhaltensänderung ist. Es wird angenommen, dass Komponenten existieren, die das Verhalten des Modells im Vergleich zu den restlichen Komponenten wenig beeinflussen. Um solche Komponenten zu finden, wird sowohl für die Weglassung in linearen als auch in parallelen Strukturen ein Quotient $0 \leq q \leq 1$ eingeführt. Je kleiner dieser ist, desto geringer ist die zu erwartende Verhaltensabweichung. Von der Regelung der Vereinfachung wird ein Grenzquotient $q_{Grenz} < 1$ festgelegt, der die Menge an wegzulassenden Komponenten begrenzt, da nur Komponenten weggelassen werden, für die gilt: $q(v) \leq q_{Grenz}$.

Regelsatz der Weglassung

Weglassungsentscheidungen können einzelne Komponenten in einer Linie betreffen oder komplette Linien in einer parallelen Linie. In Linien können nur FiFo-Puffer, Förderbänder und normale Maschinen weggelassen werden. Alle anderen Komponenten nehmen eine Umsortierung des Materialflusses vor oder bilden strukturelle Verzweigungen oder Zusammenfassungen. In parallelen Linien können komplette Linien, unabhängig von den in ihnen befindlichen Komponenten, weggelassen werden.

Lineare Strukturen

Eine wesentliche Funktion von Puffern in Fertigungsprozessen ist die Entkopplung von Maschinen. Durch die Bereitstellung von mehr Pufferplätzen kann der Durchsatz von Fertigungsprozessen erhöht werden, aber es wird häufig auch gleichzeitig der Umlaufbestand und die Durchlaufzeit vergrößert [RNT03]. Durch den Einsatz von Puffern wird die Auslastung von Engstellen erhöht, indem Wartezeiten der Maschinen durch Blockieren und Hungern reduziert werden. Blockieren wird verhindert, indem mehr Raum zum Ablegen von bearbeitetem Material vorhanden ist, und Hungern wird verhindert, indem mehr Material vor der Maschine warten kann.

Sind Maschinen im Ausgangsmodell entkoppelt, so ist die zu erwartende Verhaltensabweichung durch eine Weglassung eines Puffers groß, wenn die Maschinen dann nicht mehr

entkoppelt sind. Wenn wie von Roser et al. [RNT03] definiert wird, dass sich ein Puffer immer zwischen zwei, und nur zwei, Maschinen befindet¹⁵, dann kann ein Puffer nur dann weggelassen werden, wenn eine Entkopplung der vor und nach gelagerten Maschine weiterhin gegeben ist. Dies kann nur der Fall sein, wenn weitere Komponenten mit Puffercharakter, wie Puffer, Förderbänder und Sequenzer, zwischen den zwei Maschinen vorhanden sind, aber auch nur dann, wenn die Puffergröße der übrig bleibenden Komponenten groß genug für eine Entkopplung ist.

Um hierzu eine Aussage treffen zu können, müsste die zur Entkopplung nötige Puffergröße zwischen zwei Maschinen berechnet werden. Eine analytische Lösung dieses Problems ist möglich (bspw. [SMM07]), aber nur in Spezialfällen. Meist wird deshalb die Lösung simulativ bestimmt [SMM07, RNT03]. Deshalb soll hier mit einer einfachen Näherungslösung gearbeitet werden. Die Puffergröße der „Puffer“ zwischen den Maschinen M1 und M2 soll zur Entkopplung als ausreichend betrachtet werden, wenn sie mindestens $\max(LS(M1), LS(M2))$ entspricht. Die Wahrscheinlichkeit, dass es zum Blockieren kommt, ist gering¹⁶, weil das bearbeitete Los von M1 abgegeben werden kann. Gleiches gilt für das Hungern, weil ein komplettes Los vor M2 warten kann.

$q_{PF}(v)$ ergibt sich dann als:

$$q_{PF}(v) = \begin{cases} 0, & \text{falls } (\sum_{V_{PM}} BS(\cdot)) - BS(v) \geq \max(LS(M1), LS(M2)) \\ 1, & \text{sonst} \end{cases}$$

mit V_{PM} als Menge aller Komponenten mit Puffercharakter zwischen M1 und M2. Da $q_{PF}(v)$ auf einer sehr einfachen Näherungslösung basiert und eine abgestufte Vorhersage der Auswirkung auf die Verhaltensabweichung nicht möglich erscheint, ist $q_{PF}(v)$ eine Binärvariable. Dies bedeutet, dass Puffer uneingeschränkt weggelassen werden können oder bestehen bleiben müssen.

Maschinen können weggelassen werden, wenn sie einen unwesentlichen Engstellen-Charakter haben (Einfluss auf Durchsatz, Umlaufbestand und Durchlaufzeit) und durch ihre Bearbeitungs- und Rüstzeiten keinen großen Einfluss auf die Durchlaufzeit haben. Der Engstellen-Charakter wird festgelegt durch $C_{eff,rel}(v)$. Der Einfluss auf die Durchlaufzeit

¹⁵Liegt der Puffer vor einem Verteiler oder hinter einem Zusammenführer, dann ist diese Annahme nicht erfüllt. In diesem Fall wird approximierend eine virtuelle Maschine definiert, mit der Parametrierung, wie sie durch eine Zusammenfassung einer parallelen Linie entstehen würde.

¹⁶Die Wahrscheinlichkeit ist gering, wenn nur geringe Schwankungen im Ein- und Ausgang der Maschinen ausgeglichen werden müssen, größere Schwankungen, wie Ausfälle, können mit dieser Puffergröße nicht ausgeglichen werden.

kann abgeschätzt werden, indem die nominelle Durchlaufzeit, d.h. die Summe aller Mittelwerte der Transport-, Rüst- und Bearbeitungszeiten, des kürzesten (zeitlich gesehen) aller über die untersuchte Komponente gehender Zyklen ins Verhältnis gesetzt wird zur Rüst- und Bearbeitungszeit der untersuchten Komponente. Wie in Abschnitt 5.5.2 dargestellt, ist die Berechnung aller Zyklen im Modell aufwendig und soll vermieden werden. Zur Bestimmung des Einflusses auf die Durchlaufzeit kann, wenn der Modellgraph keine Zyklen enthält, vereinfachend der kürzeste Weg zur Supersenke und -quelle berechnet werden. Dazu werden die Transport-, Rüst- und Bearbeitungszeiten als Länge der Kanten zwischen den Komponenten verwendet. $q_{MN}(v)$ ergibt sich dann als:

$$q_{MN}(v) = \max \left(C_{eff,rel}(v), \frac{t_U(v) + t_R(v)}{t_{CT}(v)} \right)$$

mit $t_{CT}(v)$ als nominelle Durchlaufzeit des kürzesten Zyklus über v . Bei der Weglassung von Maschinen muss zusätzlich beachtet werden, dass eine Weglassung unabhängig vom Quotienten nur erfolgen kann, wenn die Maschine keinen Wert für mb gesetzt hat. In diesem Fall wäre die Validität der Simulation in Gefahr.

Förderbänder können als Puffer eingesetzt werden, können eine Engstelle darstellen und haben Einfluss auf die Durchlaufzeit. Deshalb ergibt sich $q_F(v)$ als:

$$q_F(v) = \max(q_{PF}(v), q_{MN}(v))$$

Parallele Strukturen

In parallelen Strukturen können Linien weggelassen werden, wenn ein VV-Verteiler verwendet wird und die Wahrscheinlichkeit für einen Ausgang ch (Channel, s. Abschnitt 5.2.2) gering ist und somit die an einen solchen Ausgang angeschlossene Linie sehr selten benutzt wird. $q_{VV}(ch)$ ergibt sich dann als:

$$q_{VV}(ch) = \frac{\text{Wahrscheinlichkeit für } ch}{\sum_{Ch(v)} \text{Wahrscheinlichkeit für Ausgang}}$$

Ist $q_{VV}(ch) \leq q_{Grenz}$, dann wird die Linie weggelassen, die an den Ausgang ch angeschlossen ist. Nach der Weglassung muss eine Aufteilung der Wahrscheinlichkeit des weggelassenen Ausgangs auf die anderen Ausgänge erfolgen.

Modellanalyse

Die Modellanalyse der Weglassung ist recht einfach und soll mit folgendem Algorithmus erfolgen.

Algorithmus 3 (Modellanalyse Weglassung).

```

1: function ANALYSEWEGGLASSUNG( $b, q_{Grenz}$ )
2:   Bestimmung von  $q(v)$ ,  $\forall v \in b$ 
3:    $q_{min} = \min(q(v), \forall v \in b)$ 
4:   if  $q_{min} \leq q_{Grenz}$  then
5:     return  $v$ 
6:   else
7:     return null
8:   end if
9: end function

```

Für alle Komponenten eines Teilmodells wird $q(v)$ berechnet. Wenn der kleinste Wert aller Komponenten im Teilmodell kleiner ist als q_{Grenz} , dann kann die Komponente mit diesem Wert weggelassen werden. Da sich die q -Werte der Komponenten nach jeder Weglassung ändern können, müssen die $q(v)$ vor jeder Weglassungsentscheidung neu berechnet werden.

Modellsynthese

Die Modellsynthese soll mit folgendem Algorithmus erfolgen:

Algorithmus 4 (Modellsynthese Weglassung).

```

1: procedure SYNTHESEWEGGLASSUNG( $b, v$ )
2:   if  $d(v) = VV$  then
3:     Entferne Linie mit geringster Ausgangswahrscheinlichkeit
4:     if Anzahl Ausgänge  $> 1$  then
5:       Addiere Auswahlwahrscheinlichkeit der weggelassenen Linie gleichmäßig auf
       übrige Ausgänge
6:     else
7:       Entferne  $v$  aus  $b$ 

```

```

8:      Verbinde Vorgänger von v mit Nachfolger von v
9:      z = Zusammenführer zugehörig zu v
10:     Entferne z aus b
11:     Verbinde Vorgänger von z mit Nachfolger von z
12:   end if
13: else
14:   Entferne v aus b
15:   Verbinde Vorgänger von v mit Nachfolger von v
16: end if
17: end procedure

```

Handelt es sich bei der wegzulassenden Komponente nicht um einen Verteiler, dann wird die Komponente aus dem Teilmodell entfernt und die Leerstelle geflickt. Im anderen Fall wird die Linie am Ausgang mit der geringsten Auswahlwahrscheinlichkeit weggelassen. Die Auswahlwahrscheinlichkeit wird dann auf die übrigen Ausgänge verteilt. Existiert nach der Weglassung nur noch eine am Verteiler angeschlossene Linie, dann kann auch der Verteiler und die zugehörige Zusammenfassung weggelassen werden.

5.5 Konzeption von Regelungstechniken der Vereinfachung

In diesem Abschnitt soll die Regelung der Vereinfachung konzipiert werden. Wie in Abschnitt 2.1.5 dargestellt, sollen die Regelgrößen die Komplexität der erzeugten Teilmodelle und deren Verhaltensabweichung zum Ausgangsmodell sein. Die Führungsgrößen sollen Zielvorgaben für eben diese beiden Kennzahlen sein. Als erstes soll deshalb definiert werden, wie diese Zielvorgaben quantitativ festgelegt werden können. Darauf folgend werden Methoden definiert, die festlegen, wie die Komplexität und die Verhaltensabweichung gemessen werden sollen. Methoden, mit denen die Vereinfachung beeinflusst werden kann, wurden schon in den Abschnitten 5.4.1 und 5.4.3 definiert und in die Modellanalysen und -synthesen integriert. Für die Zusammenfassung ist dies die Anzahl zu erhaltener Engstellen, also implizit die Länge zusammenzufassender Strukturen, und für die Weglassung der Quotient $q(v)$. Abschließend werden die oben genannten Kennzahlen, Größen und Methoden in einem Algorithmus der Regelung vereint.

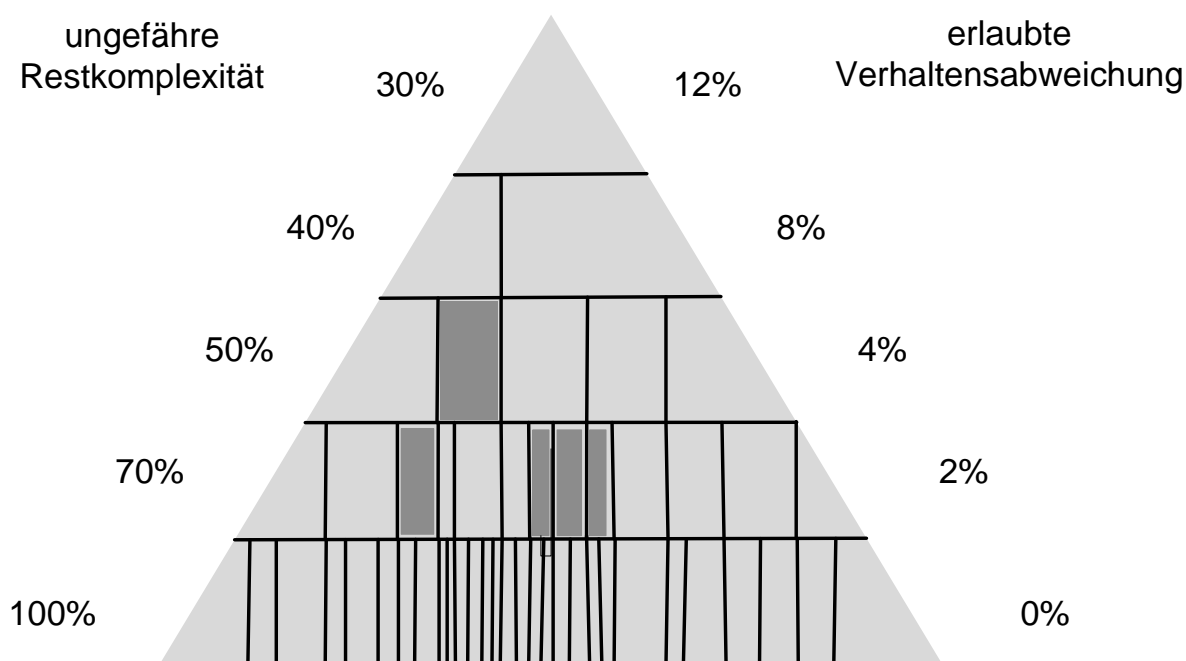


Abbildung 5.6: Zielvorgaben der Vereinfachung (Beispiel)

5.5.1 Erstellung der Zielvorgaben

Nach der Partitionierung des Ausgangsmodells existiert eine Hierarchie, die vorgibt, welche Teile des Ausgangsmodells von zu erzeugenden Teilmodellen abgebildet werden sollen. Der Benutzer gibt vor der Vereinfachung vor, wie stark die Verhaltensabweichung auf jeder der erzeugten Hierarchieebenen sein darf und wie hoch die Restkomplexität der Teilmodelle auf diesen Ebenen sein soll. Der Benutzer muss allerdings beachten, dass die Rekombination von Teilmodellen während der Laufzeit nur praktikabel eingesetzt werden kann, wenn sich die Komplexitäten von Teilmodellen benachbarter Ebenen signifikant unterscheiden. Da durch einen Rekombinationsvorgang und die nötige Zustandsgenerierung Fehler in die Simulation eingebracht werden, soll eine Rekombination nur erfolgen, wenn die Komplexität der Gesamtsimulation signifikant beeinflusst wird. Es ergibt sich also eine der Hierarchie zugeordnete Menge an Vorgaben: die Zielkomplexität $C_Z(h)$ und die einzuhaltende Verhaltensabweichung $V_Z(h)$ (beispielhaft in Abbildung 5.6 zu sehen). Die Werte müssen die Bedingungen erfüllen, dass $C_Z(h) \gg C_Z(h+1)$ und $V_Z(h) \ll V_Z(h+1)$. Wenn die Zielvorgaben wie beschrieben vorgegeben werden, existiert das Problem, dass die Zielvorgaben unter Umständen nicht erfüllt werden können. Tritt so ein Fall ein, kann also kein Teilmodell entsprechend der Vorgaben erstellt werden, so wird nur eine Ver-

knüpfung zu den Teilmodellen der Menge $B_j(b_{i,h})$ eingefügt (zu sehen an den dunklen Feldern in Abbildung 5.6). Wenn das nicht existierende Teilmodell $b_{i,h}$ aktiviert werden soll, dann wird $B_j(b_{i,h})$ aktiviert. Dies bedeutet, dass einige Partitionen des Ausgangsmodells nicht auf allen Hierarchieebenen abgebildet werden, was allerdings keinen Einfluss auf die Gültigkeit der dvdS hat.

5.5.2 Komplexitätsmaß

Wie in Kapitel 4 beschrieben, soll ein verbessertes Komplexitätsmaß durch eine Erweiterung und Kombination der Maße von Schruben und Yücesan [SY93], Formeln 3.7 und 3.8, und von Wallace [Wal87], Formel 3.4, erstellt werden.

Bei der Formel 3.7 konnte von Schruben und Yücesan die höchste Korrelation von Laufzeit und Kennwert erzielt werden, aber ein multiplikativer statt additiver Zusammenhang zwischen Transformations- und Kontrollkomplexität, wie ihn Wallace benutzt, erscheint besser. Allerdings besagt die Betrachtung des Grades einer Komponente (Wallace) in Bezug auf Kontrollkomplexität nichts, wenn schon die Ereignisse bzw. Ereignisroutinen (Schruben und Yücesan) betrachtet werden, die, bei den in dieser Arbeit genutzten Komponentenklassen, auch die Kontrollkomplexität beinhalten. Betrachtet man statt des Grades einer Komponente Pfade oder Zyklen (Schruben und Yücesan) nicht als alleiniges Maß (Formel 3.8), sondern als Gewichtung der Komponenten, so erscheint es möglich, die Auslastung von Komponenten relativ zueinander abzuschätzen. Damit kann eine Näherung erzielt werden, wie oft die Ereignisroutinen der Komponenten während der Simulation ausgeführt werden. Die Abhängigkeit von der Anzahl ausgeführter Ereignisse und der Laufzeit der Simulation wird von Hung und Leachman [HL99] dargelegt.

Die Modellkomponenten besitzen eine Menge von Ereignisroutinen und Variablen, mit denen ein Komplexitätswert berechnet werden soll. Da jede Komponente die Instanz einer Klasse ist und sich die Anzahl und Art der Variablen und Ereignisroutinen in den Komponenten einer Klasse nicht unterscheidet, muss diese Komplexitätsberechnung nur einmal für jede Klasse durchgeführt werden. Die Komplexität einer Komponentenkategorie soll sich als wie folgt berechnen:

$$C_v(d) = \text{Anzahl Ereignisse für einen Markendurchlauf} \cdot \frac{\sum \text{Variablenkomplexität}}{10} \quad (5.9)$$

Tabelle 5.8: Verwendete \mathcal{C}_v

Komponentenklasse	\mathcal{C}_v
Maschine Normal	3,4
Maschine Montage	7,5
Maschine Demontage	10,7
FiFo Puffer	2,7
LiFo Puffer	2,7
Prio Puffer	3,7
Sequenz	5,1
Förderband	4,0
Verteiler Verteilung	3,8
Verteiler Reihum	2,4
Verteiler Beschreibung	3,4
Zusammenführer	2,2
Quelle Logisch	1,2
Quelle Intervall	2,7
Quelle Verteilung	3,0
Quelle Liste	2,7
Senke Logisch	1,2
Senke Statistik	1,4

In die Berechnung gehen somit die Anzahl Ereignisse ein, die erforderlich sind, um eine Marke vom Eintritt in die Komponente zum Austritt zu bewegen, und die Variablenkomplexität. Die Variablenkomplexität wurde so festgelegt, dass Zufallsvariablen und Strings anderthalbfaches und Listen doppeltes Gewicht im Verhältnis zu deterministischen Variablen haben. Da die Berechnung der Ereignisroutinen für die Laufzeit bedeutsamer ist als die statische Anzahl Variablen, soll die Variablenkomplexität mit dem willkürlich gewählten Faktor $\frac{1}{10}$ gewichtet werden. Für die Komplexitäten der Komponentenklassen ergeben sich nach Formel 5.9 die Werte in Tabelle 5.8. Die Komplexität \mathcal{C} eines Teilmodells ergibt sich als:

$$\mathcal{C}_b^* = \sum_{v \in V_b} \mathcal{C}_v(v) \cdot \frac{Z_v(v)}{Z} \quad (5.10)$$

Die Komplexität eines Modells wird somit bestimmt durch die Summe der Komplexität seiner Komponenten, die mit einem Zyklusquotienten $\frac{Z_v}{Z}$ gewichtet werden. Der Zyklusquotient soll die Komponenten schwerer gewichten, die wahrscheinlich eine zentrale, stark frequentierte oder insgesamt wichtige Stellung im Modell haben, angezeigt durch die Anzahl Zyklen $Z_v \leq Z$, auf denen sie liegen. Komponenten, die einen großen Zyklusquoti-

enten aufweisen, liegen auf relativ vielen Pfaden, die Marken durch das Modell nehmen können; Komponenten mit einem kleinen Zyklen-Quotienten liegen auf relativ wenigen¹⁷. Analog zu der Bestimmung von dem Maß in Formel 3.8 von Schruben und Yücesan können durch Einfügen und Verbinden einer Superquelle und -senke mit dem Algorithmus nach Tarjan [VAD⁺08] alle Zyklen berechnet werden. Sind alle Zyklen ermittelt, kann bestimmt werden, auf wie vielen jede Komponente liegt. Diese Berechnung hat allerdings eine sehr lange Laufzeit und benötigt sehr viel Speicher, da die Anzahl Zyklen schon bei kleinen Modellen sehr groß sein kann und mit jedem weiteren Verteiler, Zusammenführer, Montagemaschine und Demontagemaschine sehr stark wächst.

Da die Identifizierung expliziter Zyklen kein Ziel ist, sondern nur Bestimmung der Anzahl Zyklen im Graphen Z und der Anzahl, auf denen eine Komponente liegt Z_v , kann dieses Problem mit einem wesentlich effizienteren Algorithmus gelöst werden¹⁸.

Dieser Algorithmus verwendet (auf einer separaten Datenstruktur) zuerst eine Breitensuche über alle Komponenten ausgehend von der Superquelle, um eine vorläufige Anzahl Pfade zu berechnen, und dann eine Breitensuche über alle Komponenten ausgehend von der Supersenke zur endgültigen Bestimmung der Anzahl Pfade. Jede Komponente bekommt einen Wert für die Anzahl Pfade $ap(v)$ und jede Kante (v, w) bekommt einen Wert zur Weiterreichung der Pfadanzahlen $wp(v, w)$, der mit dem Wert 1 initialisiert wird. Bei dem vorwärtsgerichteten Durchlauf des Graphen ist $ap(v) = \left(\sum_{\forall(w,v)} wp(w, v) \right) \cdot d_G^+(v)$ die Summe der eingehenden Kanten mal den Ausgangsgrad. Die wp der ausgehenden Kanten berechnen sich als $wp(v, w) = \frac{ap(v)}{d_G^+(v)}$, $\forall(v, w)$. Die endgültige Anzahl Pfade wird bei einer anschließenden Breitensuche beginnend bei der Supersenke zur Superquelle durchgeführt. Bei dem rückwärtsgerichteten Durchlauf des Graphen ist $ap(v) = \sum_{\forall(v,w)} wp(v, w)$ oder $ap(v) = \sum_{\forall(w,v)} wp(w, v)$ im Falle der Supersenke, die keine ausgehenden Kanten besitzt. Die wp werden berechnet als $wp(w, v) = \frac{ap(v)}{\sum_{\forall(w,v)} wp(w, v) \cdot wp(w, v)}$, $\forall(w, v)$.

Anbauteile werden in Montagemaschinen in einer angeschlossenen Senke vernichtet, wie in Abschnitt 5.2.1 dargestellt. Die Anbauteile bilden gewissermaßen einen eigenen Mate-

¹⁷Beispiel: Eine lineare Struktur verbunden mit einer parallelen mit vier Linien, aufgespannt durch einen Verteiler Reihum. Es existieren 4 Zyklen in diesem Modell. Alle Komponenten der linearen Struktur haben einen Zyklen-Quotienten von 1, weil sie jeweils auf 4 von 4 Zyklen liegen. Die Komponenten der parallelen Linie haben einen Zyklen-Quotienten von 0,25. Fließen Marken durch das Modell, so belegt jede Marke jede Komponente der linearen Struktur, aber nur Komponenten aus einer der vier Linien der parallelen Struktur. Die Komponenten in der linearen Struktur haben somit, wie es auch der Zyklen-Quotient angibt, eine stärkere Frequentierung als die Komponenten der parallelen Struktur.

¹⁸Der Algorithmus erfordert es, dass keine Zyklen im Graph vorliegen, es sei denn, sie werden durch Verbinden der Supersenke mit der Superquelle erzeugt.

rialfluss, da sie von ihrer Quelle bis zur Senke einer Montagestation fließen und in der Montagemaschine nicht dem Fluss der Fertigteile folgen. Deshalb wird der Graph (in der separaten Datenstruktur) so verändert, dass die Montagemaschine keine Verbindung zu den Anbauteilen und der Senke hat. Die Komponenten, von denen Anbauteile zur Montagemaschine fließen, werden direkt mit der Senke verbunden. In Demontagemaschinen wird der Materialfluss nicht wie in Verteilern aufgeteilt, sondern es werden neue Marken erzeugt, diese Tatsache soll berücksichtigt werden, indem der Graph analog verändert wird. Alle Komponenten, die neu erzeugte Marken von der Demontagemaschine aufnehmen, werden von der Demontagemaschine getrennt und direkt mit der Quelle verbunden. Die Demontagestation wiederum wird von der angeschlossenen Quelle getrennt. Durch diese beiden Graphveränderungen wäre der Graph ohne Superquelle und -senke nicht mehr komplett zusammenhängend und es entstehen mehrere Blöcke. Die Graphveränderung soll genutzt werden, um die Bezugsbasis Z zu verändern, damit eine bessere Gewichtung der Komponenten erfolgen kann. Zur Veranschaulichung der Problematik soll ein kleines Beispiel dienen: Angenommen ein Graph mit sehr vielen Verteilern und Zusammenführern, demnach auch vielen Zyklen, beinhaltet nur eine zentrale Montagemaschine, bei der an jedes Teil in Fertigteilrichtung ein Anbauteil montiert wird. Bei unveränderter Bezugsbasis würden die Komponenten der Linie des Anbauteils, über die angenommen nur ein Zyklus läuft $Z_v(\text{Anbau}) = 1$, mit dem Zyklenquotienten $\frac{Z_v(\text{Anbau})}{Z} \approx 0$ sehr gering bewertet. Da aber die Montagemaschine nur arbeitet, wenn auch Anbauteile vorhanden sind, ist diese Bewertung der Wichtigkeit der Linie der Anbauteile unzutreffend. Deshalb wird die Bezugsbasis $Z_{blc}(v)$ eingeführt, als die Anzahl Zyklen in dem Block, in dem v liegt. Zurückkommend auf obiges Beispiel ist $Z_{blc}(\text{Anbau}) = 1$ und somit $\frac{Z_v(\text{Anbau})}{Z_{blc}(\text{Anbau})} = 1$.

Die endgültige Komplexität eines Modells ergibt sich damit als:

$$C_b = \sum_{v \in V_b} C_v(v) \cdot \frac{Z_v(v)}{Z_{blc}(v)} \quad (5.11)$$

5.5.3 Messung Verhaltensabweichung

Zur Messung der Verhaltensabweichung soll das Verhalten des Ausgangsmodells mit dem der generierten Teilmodelle verglichen werden. Für ein spezielles Teilmodell ist nicht das Verhalten des gesamten Ausgangsmodells relevant, sondern nur der Partitionen, die den gleichen Teil des Systems abbilden wie das untersuchte Teilmodell. Da alle Partitionen

des Ausgangsmodells vereinfacht werden sollen, kann das Ausgangsmodell als Gesamtheit simuliert werden. Die Kennzahlen für die Verhaltensmessung und die Eingangsdaten für die Simulation der Teilmodelle können an den Partitionsgrenzen abgegriffen werden. Die Eingangsdaten für diese Teilmodell-Simulation sind die Marken, die die entsprechende Partition des Ausgangsmodells während der Simulation erhalten hat. Das Ausgangsmodell muss nur „einmal“ simuliert werden, weil sich sein Verhalten im Laufe der Vereinfachung nicht ändert. Die Teilmodelle müssen hingegen immer simuliert werden, wenn eine Verhaltensmessung erfolgen soll.

Das Verhalten eines Modells ist abhängig von seinen Komponenten, den Verknüpfungen, vom Produktionsprogramm, den Parametern der Komponenten und den verwendeten Zufallszahlen. Die Zufallszahlen bestimmen, wann eine Maschine ausfällt, wie lange eine Bearbeitung dauert, etc. Um das Verhalten eines Modells korrekt zu erfassen, müssen die verwendeten Zufallszahlen in mehreren Durchläufen der Simulation gezielt verändert werden¹⁹. Es sind mindestens 10 Simulationsläufe²⁰ - Replikationen - für jedes zu untersuchende Modell nötig, d.h. 10 für das Ausgangsmodell und jeweils 10 für jede Teilmodellmessung. Das Produktionsprogramm wird für die Verhaltensmessung nicht variiert, weil das Produktionsprogramm bei der Vereinfachung nicht berücksichtigt wird (vgl. auch Abschnitt 2.1.3); es sind demnach auch keine Möglichkeiten gegeben, regelnd einzugreifen.

Zur Bestimmung der Verhaltensabweichung sollen folgende Kennzahlen in den zu generierenden Teilmodellen und im Ausgangsmodell aufgezeichnet werden:

- Durchlaufzeit
- Umlaufbestand
- Durchsatz

Durch das Messen des Verhaltens anhand von Kennzahlen, wird die Anforderung 15 erfüllt, indem gewährleistet wird, dass das Verhalten durch eine kardinal skalierende Messung erfasst wird.

¹⁹Dies geschieht, indem den Zufallsvariablen unterschiedliche Zufallszahlenströme zugeordnet werden, aus denen die Zufallszahlen der Variable entnommen werden [LK00].

²⁰Nach Law und Kelton [LK00] sind in einem „best practice“-Ansatz 10 bis 20 Replikationen notwendig, um eine hinreichend gute Genauigkeit zu erreichen.

Bestimmung der Durchlaufzeit (LT)

Die Durchlaufzeit ist eine wichtige Kennzahl, die in vielen Simulationsexperimenten optimiert werden soll. Im Bereich Modellvereinfachung ist sie die wichtigste Kennzahl die von Rose [Ros00], Hung und Leachman [HL99] und Johnson et al. [JFM05] zur Verhaltensmessung verwendet wird. Mit der Überprüfung der Durchlaufzeit soll festgestellt werden, ob alle zeitverzögernden Variablen und Puffergrößen korrekt parametrisiert wurden und ob die Steuerstrategien stimmen. Die Durchlaufzeit wird gemessen, indem jeder Marke beim Eintritt in das Teilmodell ein Zeitstempel aufgedrückt wird. Die Differenz zu dem Zeitstempel beim Verlassen des Teilmodells ergibt die Durchlaufzeit.

Bestimmung des Umlaufbestands (WIP)

Der Umlaufbestand oder WIP ist ebenfalls eine Kennzahl, die häufig durch Simulationsexperimente optimiert werden soll. Sie wird im Bereich Modellvereinfachung nur bei Brooks und Tobias [BT00] betrachtet, und auch dort nur peripher. Der Umlaufbestand soll gemessen werden, um zu überprüfen, ob die Puffergrößen von Förderbändern und Puffern korrekt übernommen wurden. Der Umlaufbestand wird gemessen, indem die Differenz berechnet wird zwischen der Anzahl in das Teilmodell eingetretener Marken und der Anzahl ausgetretener Marken.

Bestimmung des Durchsatzes (TP)

Mit Hilfe des Durchsatzes soll überprüft werden, ob der oder die Engpässe richtig übernommen und parametrisiert wurden. Weicht die effektive Kapazität der Engpässe in der Vereinfachung von denen im Ausgangsmodell ab, so existiert eine Abweichung im Durchsatz über der Zeit. Zusätzlich kann überprüft werden, ob sich die Regeln von Verteilungskomponenten verändert haben, wenn eine Partition mehrere Ausgänge besitzt und eine separate Auswertung für jeden Ausgang vorgenommen wird. Der Durchsatz wird gemessen, indem die Anzahl aus dem Teilmodell ausgetretener Marken bestimmt wird.

Kennzahlbildung

Die Verhaltensabweichung soll für i Replikationen wie folgt bestimmt werden:

$$0 \leq \mathcal{V}(b) = \frac{\bar{\mathcal{V}}_{WIP}(b) + \bar{\mathcal{V}}_{LT}(b) + \bar{\mathcal{V}}_{TP}(b)}{3}$$

$$\begin{aligned}\mathcal{V}_{WIP,i}(b) &= \frac{\int_t |WIP_{b,i}(t) - WIP_{B_{\mathcal{H}}(b),i}(t)| dt}{\int_t WIP_{B_{\mathcal{H}}(b),i}(t) dt} \\ \mathcal{V}_{LT,i}(b) &= \frac{\int_t |LT_{b,i}(t) - LT_{B_{\mathcal{H}}(b),i}(t)| dt}{\int_t LT_{B_{\mathcal{H}}(b),i}(t) dt} \\ \mathcal{V}_{TP,i}(b) &= \frac{\int_t |TP_{b,i}(t) - TP_{B_{\mathcal{H}}(b),i}(t)| dt}{\int_t TP_{B_{\mathcal{H}}(b),i}(t) dt}\end{aligned}$$

Die Verhaltensabweichung wird als Differenzfläche zwischen den Trajektorien der Kennzahlen berechnet. Die Trajektorien werden gebildet durch die Simulation des zu untersuchenden Teilmodells b und den entsprechenden Teil des Ausgangsmodells $B_{\mathcal{H}}(b)$. Damit die Verhaltensabweichung mit zunehmender Simulationszeit nicht stetig wächst, wird die Differenzfläche auf die Gesamtfläche unter der Trajektorie des Ausgangsmodells bezogen. Es ergibt sich der relative Fehler der Kennzahlenentwicklung über der Zeit. Daraufhin wird ein Mittel der Fehler über alle Läufe erstellt $\bar{\mathcal{V}}(b)$ und schlussendlich ein Mittel über die mittleren Fehler aller Kennzahlen. Durch die Verwendung von Trajektorien zur Fehler-, bzw. Verhaltensabweichungsbestimmung wird verhindert, dass sich Fehler im Untersuchungszeitraum herauskompensieren können, wenn bspw. die Durchlaufzeit zu Beginn zu klein und gegen Ende um denselben Betrag zu groß ist. Zudem kann das Verhalten und der Verhaltensunterschied der Modelle im Zeitverlauf durch die Verwendung von Trajektorien gut visualisiert werden und der vergleichende Charakter der Messung wird betont (trace-driven Validierung s. Abschnitt 3.4.2).

5.5.4 Algorithmus der Regelung

Nun soll der Algorithmus entwickelt werden, der die Modellanalyse und Synthese der beiden Vereinfachungsmethoden Zusammenfassung (Abschnitt 5.4.1) und Weglassung (Abschnitt 5.4.3) auf eine Weise steuert, dass die Zielvorgaben der Vereinfachung erfüllt werden. Kerngedanke des Algorithmus 5 soll es sein, dass die Anzahl an Verhaltensmessungen klein gehalten werden muss, da sie sehr zeitaufwendig sind. Die Komplexitätsmessung hingegen kann sehr schnell ausgeführt werden und kann deshalb wesentlich häufiger erfolgen. Durch den Algorithmus der Regelung soll ein wie in Abbildung 5.7 skizzierter idealer Verlauf der Vereinfachung erreicht werden. Zuerst wird versucht, die Komplexität zu senken, ohne das Verhalten stark zu beeinflussen (große negative Steigung der Kurve). Bis die Zielkomplexität erreicht ist, wird eine immer höhere Verhaltensabweichung akzeptiert (abnehmende negative Steigung der Kurve). Wurde die Zielkomplexität erreicht,

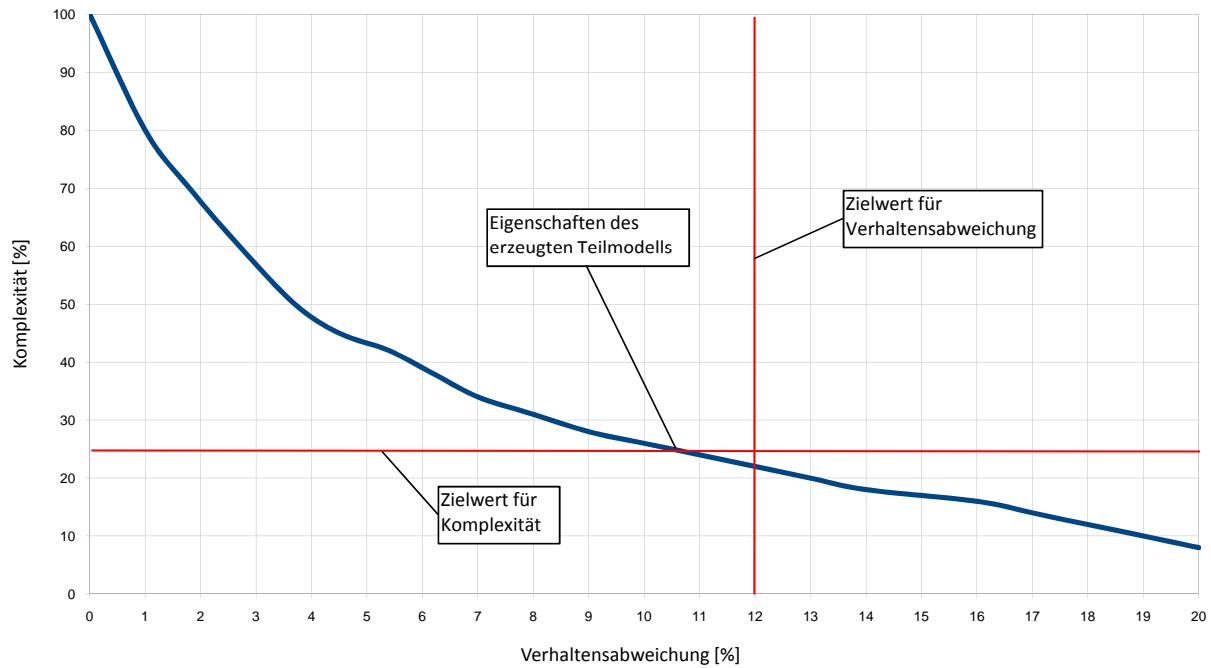


Abbildung 5.7: Komplexitäts- Verhaltensabweichungsverlauf für eine Vereinfachung (Skizze)

findet eine Verhaltensmessung statt. Attestiert diese dem erzeugten Teilmodell eine zu große Verhaltensabweichung, dann wird in einem letzten Schritt geprüft, ob eine Nicht-Berücksichtigung der Rüstzeiten bei der Zusammenfassung Besserung bringt. Ist dieses auch nicht der Fall, dann wird für die zu vereinfachende Partition kein Teilmodell in die Hierarchie eingefügt.

Algorithmus 5 (Regelung der Vereinfachungsverfahren).

Require: $b = B_j(b_h), \mathcal{C}_Z(h), \mathcal{V}_Z(h), \vartheta_{BN}, \vartheta_q$

- 1: Bestimme $\mathcal{C}(b)$
- 2: $q_{Grenz} = \vartheta_q$
- 3: Liste $L_C = \{C_{eff,rel}(v) | v \in b\}$
- 4: $L_C.sort(absteigend)$
- 5: Liste $L_{BN} = \{v | L_C.indexOf(C_{eff,rel}(v)) \leq L_C.length \cdot \vartheta_{BN}\}$
- 6: $L_{BN}.sort(absteigend)$
- 7: **while** $L_{BN}.length > 1$ **do**
- 8: Struktur $s = \text{ANALYSEZUSAMMENFASSUNG}(b, L_{BN})$
- 9: **while** $\mathcal{C}(b) > \mathcal{C}_Z(h)$ und $s \neq null$ **do**
- 10: $\text{SYNTHESEZUSAMMENFASSUNG}(b, s, L_C)$

```

11:     Bestimme  $\mathcal{C}(b)$ 
12:      $s = \text{ANALYSEZUSAMMENFASSUNG}(b, L_{BN})$ 
13:      $L_C = \{C_{eff,rel}(v) | v \in b\}$ 
14:     end while
15:     Komponente  $k = \text{ANALYSEWEGGLASSUNG}(b, q_{Grenz})$ 
16:     while  $\mathcal{C}(b) > \mathcal{C}_Z(h)$  und  $k \neq \text{null}$  do
17:          $\text{SYNTHESEWEGGLASSUNG}(b, k)$ 
18:         Bestimme  $\mathcal{C}(b)$ 
19:          $k = \text{ANALYSEWEGGLASSUNG}(b, q_{Grenz})$ 
20:     end while
21:     if  $\mathcal{C}(b) > \mathcal{C}_Z(h)$  then
22:          $L_{BN}.remove(\text{letztes Element})$ 
23:          $q_{Grenz} += \vartheta_q$ 
24:     end if
25: end while
26: if  $\mathcal{C}(b) \leq \mathcal{C}_Z(h)$  then
27:     Bestimme  $\mathcal{V}(b)$ 
28:     if  $\mathcal{V}(b) \leq \mathcal{V}_Z(h)$  then
29:         Einfügen des Teilmodells in Hierarchie
30:     else
31:         Schalte Berücksichtigung von Rüstzeiten aus
32:         Bestimme  $\mathcal{V}(b)$ 
33:         if  $\mathcal{V}(b) \leq \mathcal{V}_Z(h)$  then
34:             Einfügen des Teilmodells in Hierarchie
35:         else
36:             Teilmodell konnte nicht erzeugt werden
37:         end if
38:     end if
39: else
40:     Teilmodell konnte nicht erzeugt werden
41: end if

```

Eingaben für diesen Algorithmus sind das Teilmodell b , welches aus den zugehörigen Teilmodellen der Hierarchieebene $j = h + 1$ zusammengesetzt wird, die Zielkomplexität, die Zielverhaltensabweichung, ein Engstellenerhaltungsquotient $0 < \vartheta_{BN} < 1$ und

ein q_{Grenz} -Inkrement ϑ_q . ϑ_{BN} legt fest, wie viele Komponenten aus b zu Beginn als zu erhaltene Engstellen festgelegt werden sollen (Zeile 5). q_{Grenz} wird kontinuierlich um ϑ_q erhöht, damit q_{Grenz} nicht zu groß wird, muss darauf geachtet werden, dass ϑ_q im richtigen Verhältnis zu ϑ_{BN} steht (Zeile 22 und 23). In den Zeilen 8, 10, 15 und 17 werden die Algorithmen 1, 2, 3 und 4 aus Abschnitt 5.4 aufgerufen.

Die Auswirkungen der Vereinfachung auf die Verhaltensabweichung wird durch die Anzahl zu erhaltener Engstellen und den q -Wert bestimmt. Ist die Anzahl zu erhaltener Engstellen hoch, so wird sich das Verhalten des erzeugten Teilmodells mit hoher Wahrscheinlichkeit nicht wesentlich vom Ausgangsmodell unterscheiden, aber die erreichbare Komplexitätsreduzierung ist stark begrenzt. Mit abnehmender Anzahl zu erhaltener Engstellen kann die Komplexität weiter verringert werden, aber die Auswirkungen auf das Verhalten sind wahrscheinlich höher. Aus diesem Grund wird die Anzahl zu erhaltener Engstellen kontinuierlich bis zum Erreichen der Zielkomplexität gesenkt (Zeile 22). Eine analoge Steuerung erfolgt mit dem q -Wert, dessen Grenzwert kontinuierlich erhöht wird (Zeile 23).

5.6 Konzepte zur Generierung von Zustandsabbildungsfunktionen

In diesem Abschnitt soll ein Konzept entwickelt werden, wie der Zustand der zu deaktivierenden Teilmodelle genutzt werden kann, um einen konsistenten und qualitativ guten Zustand der zu aktivierenden Teilmodelle zu erzeugen (s. Abschnitt 2.1.6).

5.6.1 Grundlagen der Zustandsgenerierung

Der Zustand eines Teilmodells ist definiert als die Zustände der Komponenten des Teilmodells (siehe Definition 18). Der Zustand einer Komponente setzt sich zusammen aus den Zustandsvariablen, der Belegung und der Liste der eingeplanten Ereignisse. Die Liste der eingeplanten Ereignisse definiert, wann welche Ereignisroutinen ausgeführt werden sollen. Diese Routinen ändern die Zustandsvariablen, die Belegung und planen Ereignisse ein. Die Zustandsvariablen können unterteilt werden in solche, die sich auf die Belegung beziehen²¹,

²¹Bspw.: Losgröße erreicht, Eingang offen, etc.

solche die sich auf die Störungen beziehen²², und solche, die Statistiken repräsentieren²³. Die Belegung ist die Markenmenge, die sich auf einer Komponente befindet.

Beim Umschalten in der dvdS werden die Teilmodelle $B_j(b_{i,h})$ deaktiviert und das Teilmodell $b_{i,h}$ aktiviert oder umgekehrt. Die Komponentenmenge, -art und -verknüpfung ist in $B_j(b_{i,h})$ und $b_{i,h}$ wegen der Vereinfachung unterschiedlich. Diese Unterschiedlichkeit steigt mit steigendem Vereinfachungsgrad. Durch eine Zustandsabbildungsfunktion

$$\psi(s(B_{deaktivieren})) \approx s(B_{aktivieren})$$

muss demnach der Zustand einer Komponentenmenge auf eine grundsätzlich andere abgebildet werden.

Die beiden Komponentenmengen sind aber nicht unabhängig voneinander, sondern es existiert eine „Erzeugungsbeziehung“ durch die Vereinfachung. Das vereinfachte Teilmodell $b_{i,h}$ ist durch eine Menge von Vereinfachungsoperationen Θ aus $B_j(b_{i,h})$ erzeugt worden. Eine Menge Komponenten V_K aus $B_j(b_{i,h})$ wird mit einer geordneten Menge von Vereinfachungsoperationen $\vec{\theta} \subseteq \Theta$ auf eine Komponente v_V aus $b_{i,h}$ abgebildet. Diese „Erzeugungsbeziehung“ kann genutzt werden, um Zustände von V_K auf v_V zu übertragen und umgekehrt.

Dazu soll wie folgt vorgegangen werden: Als erstes wird der Zustand von Komponenten direkt und unverändert übertragen, die in beiden Mengen existieren, bspw. der Engpass in einer vereinfachten Linie oder wenn gar keine Vereinfachung vorgenommen wurde. Danach wird die Markenbelegung übertragen²⁴, es werden also alle Marken von der einen Komponentenmenge entfernt und in der anderen hinzugefügt. Aus der Belegung einer Komponente ergeben sich Ereignisse, wie bspw. der Arbeitsbeginn oder der Versuch des Verschickens. Aus der Belegung folgt also eine Ereignismenge, die eingeplant wird. Zur Bestimmung der richtigen Einplanungszeitpunkte wird die Ereignisliste im zu deaktivierenden Teilmodell verwendet. Abschließend werden die Zustandsvariablen übertragen, die u.U. auch die Einplanung von Ereignissen erfordert, wie bspw. die Einplanung des Störungsendereignisses, wenn die Maschine gestört ist.

²²Bspw.: Maschine gestört

²³Bspw.: kumulierte Wartezeit, Durchsatz, etc.

²⁴Erklärendes Beispiel: Sei θ Vereinfachungsregel in Tabelle 5.4, so muss das Förderband F1 alle Marken aufnehmen, die sich im komplexen Modell auf Maschinen Mn, den Förderbändern Fn und den Puffern Pn, mit $n < z$, befinden. Soll ein komplexeres Teilmodell aktiviert werden, so gilt die Gegenrichtung analog.

In den nächsten drei Abschnitten soll dieses Grobkonzept verfeinert werden. Dabei wird zusätzlich unterschieden zwischen Zustandsvariablen, die Ereignisse erzeugen (Störungen), und solche, für die das nicht gilt (Statistiken).

5.6.2 Bestimmung der Belegung der zu aktivierenden Teilmodelle

Für einen konsistenten Zustand muss die Belegung der zu aktivierenden Komponentenmenge aus denselben Marken bestehen wie die Belegung der zu deaktivierenden. Wenn eine Marke verloren ginge, so könnte ein inkonsistenter Zustand entstehen²⁵.

Um einen qualitativ guten Zustand zu erhalten, sollen Marken zu ähnlichen Zeitpunkten aus der zu aktivierenden Komponentenmenge austreten, wie aus der zu deaktivierenden. Die Austrittszeitpunkte in der zu deaktivierenden Komponentenmenge können aus der Ereignisliste bestimmt werden. Dort ist festgelegt, welche Marken welche belegungsbedingten Ereignisse ausgelöst haben. Es kann somit für jede Marke ein ungefährer Austrittszeitpunkt aus der zu deaktivierenden Komponentenmenge berechnet werden, indem zu dem Austrittszeitpunkt aus der Komponente, auf der sich die Marke befindet, die Mittelwerte der Bearbeitungs- und Förderzeiten der nachfolgenden Komponenten bis zur letzten der Menge hinzu addiert werden. Es werden nun alle Marken nach diesem Wert in aufsteigender Reihenfolge sortiert.

In der zu aktivierenden Komponentenmenge werden die Marken in dieser Reihenfolge, vom Ausgang der Menge her, auf die Komponenten verteilt. Damit die Austrittszeitpunkte eingehalten werden können, müssen die belegungsbedingten Zustände (siehe unten) in der Komponentenmenge gleich bleiben. Wenn die Komponente am Ausgang einer zu deaktivierenden Menge nur belegt ist, und somit noch kein Ausgangereignis eingeplant ist, dann kann die Ausgangskomponente in der zu aktivierenden Menge nicht voll sein, weil dann ein Ausgangereignis eingeplant werden muss. Aus der Platzierung der Marken entstehen also Ereignisse, die Ausführungszeiten dieser werden angepasst, um die Austrittszeiten der Marken aus der Komponentenmenge im Vergleich zur deaktivierten Menge einzuhalten. Die Ereignisse von Komponenten werden zu den vorher ermittelten ungefähren Austrittszeitpunkten eingeplant, abzüglich der Bearbeitungs- und Förderzeiten nachfolgender Komponenten in der Menge.

²⁵Beispiel: Ein Tisch soll aus einer Tischplatte und vier Beinen montiert werden, geht ein Bein verloren, so kann der Tisch nicht montiert werden. Existiert kein weiteres Bein im Modell, so könnte die Montagemaschine unbefristet blockieren.

Ist diese Methode für lineare Strukturen in der Komponentenmenge relativ einfach durchzuführen, so entsteht bei parallelen Strukturen ein Problem, wenn aus einer parallelen eine lineare Struktur entstanden ist. Soll die Belegung einer solchen linearen Struktur auf eine parallele abgebildet werden, so entsteht das Problem, auf welche der parallel angeordneten Komponenten die Marken gelegt werden sollen. Als Lösung kann die Verteilungswahrscheinlichkeit in den VV und VR Komponenten verwendet werden. Die Belegung wird so gewählt, dass die relative Markenmenge in jeder der parallelen Komponenten den Verteilungswahrscheinlichkeiten entspricht.

Bei der Vereinfachung durch Weglassung wird eine Komponente oder eine Komponenten-
gruppe einer Leerstelle zugeordnet. Soll ein vereinfachtes Teilmodell aktiviert werden, bei dem die Leerstelle vorliegt, dann werden die zu verteilenden Marken auf die Nachfolgekomponente gelegt. Für die andere Richtung entstehen keine Probleme, da eine Leerstelle keinen zu übertragenden belegungsbedingten Zustand haben kann.

Belegungsbedingte Zustände der Komponentenklassen

Die Grundzustände, die durch eine Belegung mit Marken entstehen können, sind: voll, belegt und leer. Diese Zustände gelten für alle Komponentenklassen. Wenn die Puffer- bzw. Losgröße einer Komponente 1 ist, so existiert der Zustand belegt nicht.

Für Maschinen gilt:

voll	Ausgabe in Restdauer Bearbeitungszeit, Inputchannel schließen
belegt	Inputchannel öffnen
leer	Inputchannel öffnen

Für Puffer gilt:

voll	Ausgabe, Inputchannel schließen
belegt	Ausgabe, Inputchannel öffnen
leer	Inputchannel öffnen

Für Sequenzer gilt:

voll	Sequenzsuche, Inputchannel schließen
belegt	Sequenzsuche, Inputchannel öffnen
leer	Inputchannel öffnen

Für Förderbänder gilt:

voll	Ausgabe nach Restdauer Förderzeit, Inputchannel schließen
belegt	Ausgabe nach Restdauer Förderzeit, Inputchannel öffnen
leer	Inputchannel öffnen

Für Verteiler und den Zusammenführer gilt:

voll	Ausgabe, Inputchannel schließen
leer	Inputchannel öffnen

Für Quellen gilt:

voll	Ausgabe
leer	Neue Marke erzeugen

Für Senken gilt:

immer	Inputchannel öffnen
-------	---------------------

Ist eine Belegung erkannt, so wird die entsprechende Ereignisroutine (Ausgabe, Sequenzsuche oder Neue Marke erzeugen) eingeplant. Das Öffnen und Schließen der Inputchannel sind keine Ereignisroutinen, sondern Methoden des Simulationskernels, die ausgeführt werden. Ist ein Inputchannel geschlossen, so kann keine weitere Marke über diesen Channel in die Komponente eintreten.

5.6.3 Bestimmung der Störstatus der zu aktivierenden Teilmodelle

Maschinen sind die einzigen Komponentenklassen, die Störeigenschaften besitzen und zum Zeitpunkt des Umschaltens gestört sein können. In der Vereinfachung durch Zusammenfassung wird in jeder Struktur mit Maschinen nur eine Maschine als Engpass übernommen, alle anderen gehen in Förderbändern auf. In der Komponentenmenge der komplexen Teilmodelle existieren somit im Allgemeinen mehr Maschinen.

Liegt zum Zeitpunkt der Deaktivierung eine Störung vor, dann werden alle der gestörten Maschine über $\vec{\theta}$ zugeordneten Maschinen im zu aktivierenden Teilmodell gestört. Im Falle, dass ein vereinfachtes Teilmodell aktiviert werden soll, wird die Stördauer des übernommenen Engpasses bestimmt als das Maximum der Reststörzeit aller gestörten zugeordneten

Maschinen im komplexen Teilmodell. Soll eine Menge an komplexeren Teilmodellen aktiviert werden, so sollen alle Maschinen gestört werden, die in vereinfachten Strukturen mit dem erhaltenen Engpass lagen, der bei der Deaktivierung gestört war. Das Störverhalten des Engpasses wird in der Vereinfachung durch Zusammenfassung so verändert, dass seine Verfügbarkeit der Gesamtverfügbarkeit aller substituierten Maschinen entspricht (siehe Abschnitt 5.4.1). Eine alleinige Störung der Engpassmaschine ist demnach nicht richtig. Eine Störung aller zugeordneten Maschinen ist ebenfalls keine richtige Lösung, sondern entspringt der Annahme des schlechtesten Falls, da eine korrekte Berechnung der gestörten Maschinen und ihrer Reststördauern nicht möglich ist. Als Stördauer wird das Minimum aus der Reststördauer des zu deaktivierenden Engpasses und der normalen Stördauer jeder Maschine eingesetzt. Für alle Komponenten, die bei der Zustandsgenerierung nicht gestört sind, wird eine nächste Störung mit den Methoden des Simulationskernels anhand ihrer Störeigenschaften berechnet. Ist eine Weglassung vorgenommen worden und sollen komplexere Teilmodelle aktiviert werden, so ist kein Störstatus zu übernehmen; in diesem Fall wird die nächste Störung ebenfalls vom Simulationskernel anhand ihrer Störeigenschaften berechnet.

5.6.4 Bestimmung der Statistikvariablen der zu aktivierenden Teilmodelle

Der Durchsatz wird für alle Komponenten im Modell berechnet. Zur Zustandsgenerierung erhält jede Komponente der zu aktivierenden Komponentenmenge den Durchsatz der Komponente am Ausgang der zu deaktivierenden Menge plus der Anzahl Marken ihrer Vorgänger. Alle Marken, die den Durchsatz der vorgelagerten Komponenten erhöhen könnten, sind im Zuge der Belegungsgenerierung auf diese verteilt worden. Ist dieses Verfahren bei linearen Strukturen fehlerlos, so würde der Durchsatz in parallelen deutliche Fehler aufweisen. Deshalb wird in parallelen Strukturen der Durchsatz jeder Komponente der Verteilungswahrscheinlichkeit des verwendeten Verteilers angepasst.

Die restlichen Statistiken sind die Auslastung, der mittlere Bestand und die kumulierten Zeiten für Arbeit, Rüsten, Stören und Warten. Für diese Variablen können im Allgemeinen keine zu übertragenden Werte berechnet werden. Wird eine Linie von Förderbändern durch ein Förderband vereinfacht oder geschieht gleiches mit Puffern, dann lassen sich diese Statistiken berechnen, weil alle beteiligten Komponenten die gleichen Statistiken

haben. In dem am häufigsten vorkommenden Fall, Förderbänder substituieren Maschinen, können keine der oben genannten Statistikvariablen übertragen werden, weil Maschinen und Förderbänder unterschiedliche Statistiken besitzen. Die einzige Überschneidung ist die Auslastung, die aber bei Förderbändern als Durchsatz bezogen auf den maximalen Durchsatz berechnet wird und bei Maschinen als Arbeits-, Rüst- und Störzeiten bezogen auf die gesamte verfügbare Zeit. Deshalb werden bei einer Umschaltung alle Zeitstatistiken und der mittlere Bestand bei nicht übernommenen Komponenten (z.B. Engstellen) gelöscht und ab dem Umschaltzeitpunkt neu berechnet.

Ist als Vereinfachungsoperation eine Weglassung vorgenommen worden und sollen komplexere Teilmodelle aktiviert werden, so erhalten Komponenten, die keine Verbindung zu Komponenten des vereinfachten Teilmodells haben, den Durchsatz ihres Nachfolgers und alle übrigen Statistikvariablen werden ab dem Umschaltzeitpunkt neu berechnet.

6 Validierung

I love it
when a plan comes together.

*(John 'Hannibal' Smith,
The A-Team)*

In diesem Kapitel sollen die in der Konzeption entworfenen Methoden und Techniken untersucht werden. Bevor eine Validierung erfolgen konnte, wurden die Methoden in der d³FACT Umgebung implementiert. Als erster Schritt der Validierung wird ein Versuchsplan erstellt, der festlegt, mit welchen Versuchen und Versuchsaufbauten die entwickelten Methoden untersucht werden sollen, um zu überprüfen, ob die Anforderungen aus Abschnitt 2.2 erfüllt werden können.

6.1 Übersicht

In Tabelle 6.1 ist eine Übersicht über alle in Abschnitt 2.2 definierten Anforderungen und die Versuche gegeben, die durchgeführt werden sollen, um zu überprüfen, ob die erstellten Methoden und Techniken die Anforderungen erfüllen. Nicht für alle Anforderungen können oder sollen separate Versuche durchgeführt werden, weil sie, als Voraussetzung für die Validität anderer Methoden und Techniken, mit diesen zusammen untersucht werden oder nicht effektiv durch Versuche validiert werden können.

Die Anforderungen 1 und 5 sind Voraussetzungen für die Vereinfachung; kann diese durchgeführt werden, sind diese Anforderungen erfüllt. Die Anforderungen 2, 3 und 4 wurden schon während der Konzeption im Abschnitt 5.2.1 diskutiert und argumentativ validiert. Anforderung 9 ist erfüllt, wenn eine valide Partitionierung erfolgt, so dass keine separaten Versuche durchgeführt werden müssen. Werden in den Versuchen zur Untersuchung

Tabelle 6.1: Versuchsplan

Nr	Bereich	Untersuchte Anforderung	Versuchsaufbau
1	Modellbeschreibung	Automatisierbare Modellanalyse	–
2	Modellbeschreibung	Erhalt der Flexibilität	–
3	Modellbeschreibung	Schlankheit der Modelle	–
4	Modellbeschreibung	Änderungen zu d ³ FACT	–
5	Modellbeschreibung	Kapselung	–
6	Partitionierung	geom. Zusammenhang	Partitionierung von generischen Modellen
7	Partitionierung	log. Zusammenhang	Partitionierung von generischen Modellen
8	Partitionierung	ausgeglichener Umfang	Partitionierung von generischen Modellen
9	Hierarchisierung	Validität	–
10	Vereinfachung	Regelbarkeit	–
11	Vereinfachung	Stufenlosigkeit	Vereinfachung mit quasi kontinuierlichen Zielvorgaben für Zielkomplexität
12	Vereinfachung	Stochastik erhalten	Vergleich der Streuung von Statistik-kennzahlen zwischen Teilmodellen unterschiedlicher Komplexitätsstufen
13	Regelung	Erfüllung der Vorgaben	Durchführung der geregelten Vereinfachung
14	Regelung	Korrelation \mathcal{C} zu \mathcal{L}	Bestimmung des Korrelationskoeffizienten von Laufzeit und Komplexität verschiedener Teilmodelle
15	Regelung	Kardinale Skalierung	–
16	Zustandsabbildung	Qualität	Vergleich von Zuständen verschiedener Teilmodelle vor und nach Umschaltung
17	Integration	Fähigkeit	–

der Regelung Teilmodelle erzeugt, die den Anforderungen entsprechen, so ist auch Anforderung 10 erfüllt. Anforderung 15 wurde in Abschnitt 5.5.3 argumentativ validiert. Da alle Versuche mit d³FACT und angeschlossenen Modulen durchgeführt werden, ist Anforderung 17 mit der Durchführung der nachfolgend beschriebenen Versuche erfüllt.

6.2 Testmodelle

Es wurden drei Test-Ausgangsmodelle erstellt, um die Versuche zur Validierung durchführen zu können. Jedes dieser Modelle hat dabei andere Struktur-, Komplexitäts- und Verhaltenseigenschaften. Modell A (siehe Abbildung 6.1, Flussrichtungen durch Pfeile gekennzeichnet) ist von hoher Komplexität und hat einen einzigen Fertigteilfluss, so dass alle Quellen außer einer einzigen Anbauteile erzeugen, die in Montagemaschinen an das Fertigteil montiert werden. Das Modell enthält fünf parallele Linien, in die der Fertigteilfluss aufgeteilt wird, zwei von diesen sind nur teilweise homogen. Durch die fünf im Modell vorhandenen Montagemaschinen und dadurch bedingt fünf Linien für Anbauteile entstehen persistente Strukturen, weil die Linien für Anbauteile zwar in sich vereinfacht werden, aber nicht wegfallen können. Die maximale Komplexitätsreduktion soll als mittelmäßig eingestuft werden. Modell B (siehe Abbildung 6.2) ist von mittlerer Komplexität, hat allerdings prozentual und absolut mehr Montagemaschinen (6). Es existieren 2 Fertigteilflüsse mit homogenen parallelen Linien und neben den kurzen Flüssen für Anbauteile auch einen langen (Modellmitte). Der lange Fluss von Anbauteilen wird in drei Teile geteilt und dient drei Montagestationen als Zuführung von Anbauteilen. Die Flüsse von Anbauteilen besitzen ebenfalls homogene parallele Linien. Durch die hohe Anzahl an Montagemaschinen und die Aufteilung des Materialflusses für Anbauteile entstehen sehr viele persistente Strukturen und die maximale Komplexitätsreduktion ist schlecht. Modell C (siehe Abbildung 6.3) ist von geringer Komplexität und besitzt eine einzige Flussrichtung für Fertigteile und keine Montagestationen. Es besteht aus einer Sequenz aus Linien und homogenen parallelen Linien und besitzt somit eine sehr gute Vereinfachungsfähigkeit.

Die Größe dieser Modelle ist für die Validierung der Methoden nicht entscheidend, weil aufgrund des iterativen Vorgehens in der Partitionierung und Vereinfachung jedes dieser Modelle auch als ein Teilmodell in einem größeren Modell aufgefasst werden kann, welches auf einer beliebigen Stufe der Partitionierung bzw. Vereinfachung erzeugt wurde.

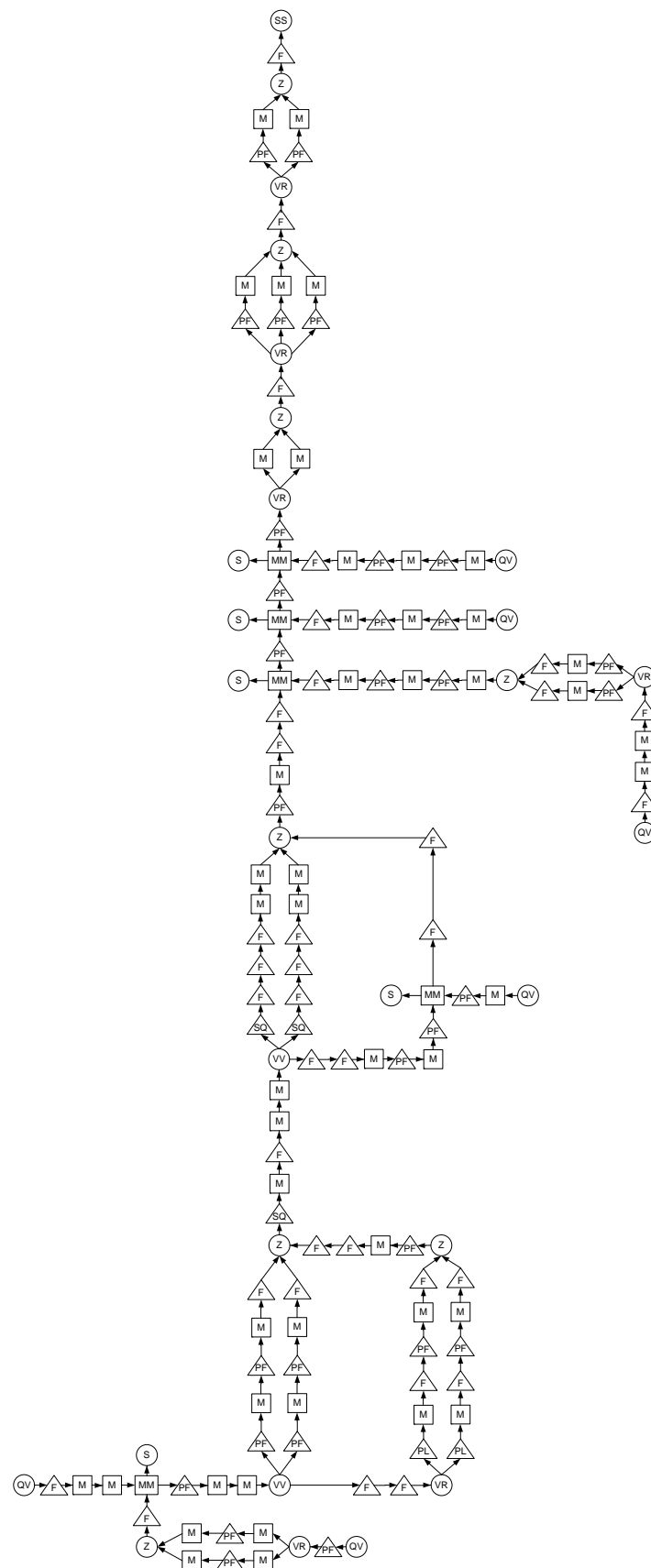


Abbildung 6.1: Struktur und Materialflussrichtung Modell A

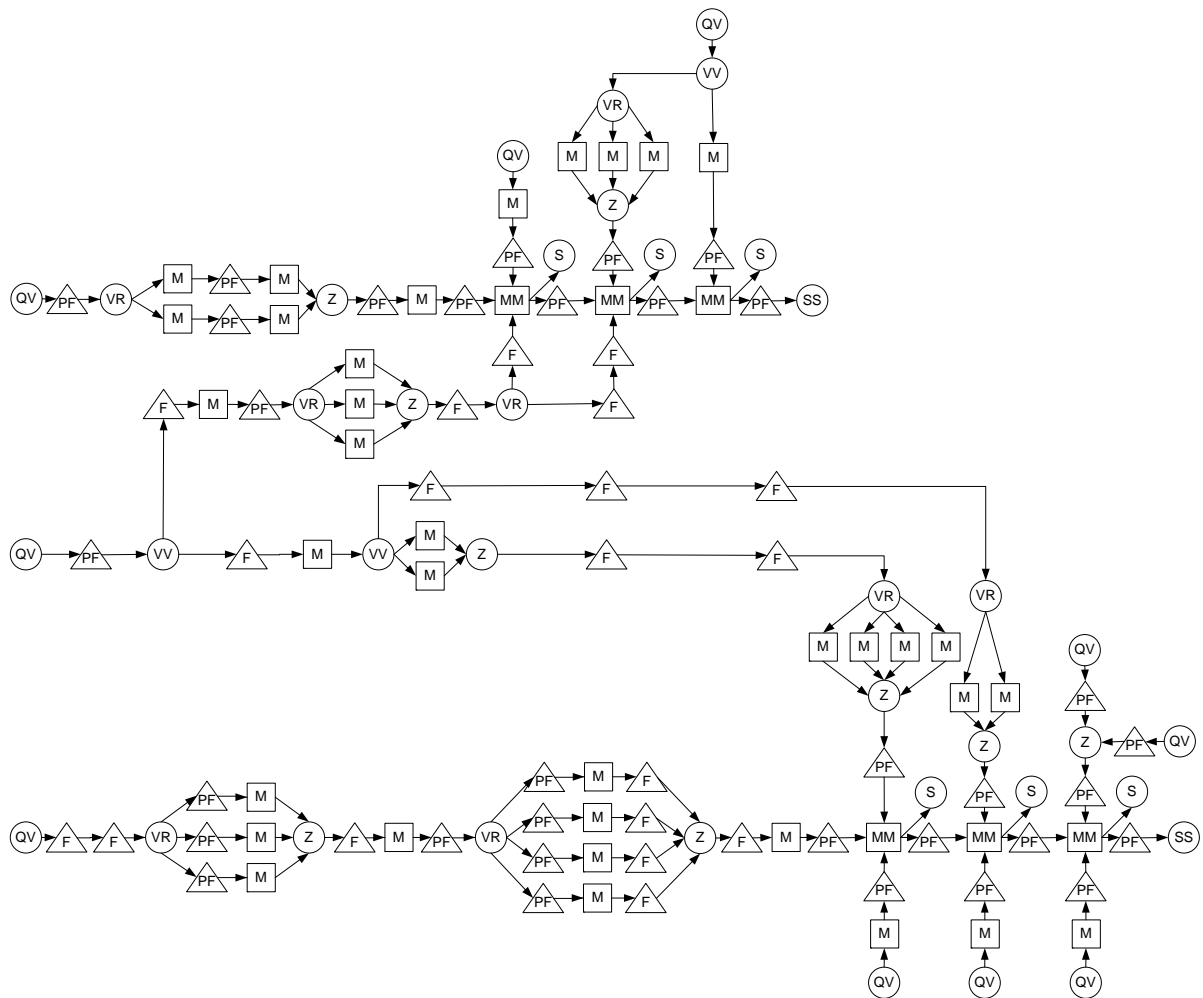


Abbildung 6.2: Struktur und Materialflussrichtung Modell B

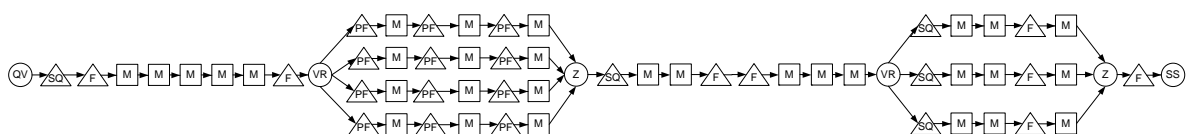


Abbildung 6.3: Struktur und Materialflussrichtung Modell C

6.3 Validierung der Partitionierungsmethode

Zur Validierung der Anforderungen 6, 7 und 8 sollen alle drei Ausgangsmodelle partitioniert werden. Jedes Modell soll zweimal partitioniert werden und zwar so, dass einmal die Partitionsanzahl relativ klein $n = 3$ und einmal relativ groß $n = 5$ ist. Damit soll untersucht werden, ob die Zusammenhangs- und Balanceanforderungen auch erfüllt werden, wenn die Zielerreichung schwieriger ist ($n = 5$). Zudem sollen beide in Kapitel 5.3 erstellten Zielfunktionen miteinander verglichen und die bessere herausgestellt werden.

Zuerst sollen optimale Werte für die Parameter der Zielfunktionen 5.2 und 5.3 BF und λ bestimmt werden. Dafür werden Partitionierungen ($n = 5$) mit allen Modellen durchgeführt und der Parameter BF im Intervall $[0, 1; 1, 9]$ und λ im Intervall $[5, 130]$ variiert. Da es sich hier um einen randomisierten Algorithmus handelt, bei dem im HC-Matching und dem Greedy-Algorithmus Knoten in unterschiedlicher Reihenfolge untersucht werden, sollen bei jedem Parameterwert 1000 unabhängige Läufe durchgeführt werden. Der optimale Parameterwert soll bestimmt werden, indem die Mittelwerte des Ratio-Cuts verglichen werden. Um den Mittelwert vor dem Einfluss von Ausreißern zu schützen, sollen die 10 größten und 10 kleinsten Ratio-Cuts weggeschnitten werden.

Um die bessere der beiden Zielfunktionen zu identifizieren, sollen die Ratio-Cut-Werte für die optimalen Parameterwerte verglichen werden. Zusätzlich zu der Größe des Ratio-Cuts soll untersucht werden, ob sich zwischen den Zielfunktionen die Streuung des Ratio-Cuts signifikant unterscheidet. Wenn sich die Größe des Ratio-Cuts nicht wesentlich unterscheidet, soll die Zielfunktion als besser gelten, die die kleinere Streuung aufweist. Ist die bessere Zielfunktion identifiziert, soll grafisch überprüft werden, ob die Anforderungen bzgl. Zusammenhang und Balance erfüllt sind. Diese Auswertung soll grafisch erfolgen, weil so sehr intuitiv die Güte der Partitionierung bewertet werden kann.

6.3.1 Ergebnisse

Die Auswertung der Versuche zur Findung der optimalen Parameterwerte für die Zielfunktionen ist in Tabelle 6.2 dargestellt. Die maximalen Werte des Ratio-Cuts ergeben sich nicht für jedes Modell bei denselben Parameterwerten. Um für jede Zielfunktion einen einzigen Wert festlegen zu können, wurden Parameterwerte gesucht, bei denen sich Ratio-Cut-Werte nahe dem Maximum eingestellt haben. Für Modell A ergeben sich sehr

Tabelle 6.2: Bestimmung des maximalen Ratio-Cuts

Modell	Zielfunktion 5.3		Zielfunktion 5.2	
	λ	rc $[10^7]$	BF	rc $[10^7]$
A	40	5,18	0,9	6,37
B	40	4,84	1	6,99
C	30	0,052	1,1	0,058

Tabelle 6.3: Variationskoeffizienten des Ratio-Cuts

Modell	Variationskoeffizient $[10^7]$	
	ZF 5.3	ZF 5.2
A	1,99	2,89
B	2,82	6,99
C	0,0069	0,0068

gute Werte für $35 \leq \lambda \leq 45$ und $0,9 \leq BF \leq 1,2$, für Modell B $30 \leq \lambda \leq 60$ und $1 \leq BF \leq 1,2$ und für Modell C $25 \leq \lambda \leq 40$ und $0,9 \leq BF \leq 1,1$. Der von Karypis und Kumar [KK98b] verwendete Wert $BF = 1$ liegt für jedes Modell im Intervall und soll auch hier als Optimum verwendet werden. Das Maximum des Ratio-Cuts stellt sich bei Modell A und B unter Verwendung von Zielfunktion 5.3 bei $\lambda = 40$ ein und dieser Parameterwert liegt auch im Bereich guter Werte von Modell C; deshalb ist das Optimum von λ der Wert 40. Die Streuung der Ergebnisse wird in Tabelle 6.3 durch den Variationskoeffizienten $V_k = \frac{\sigma}{\bar{x}}$ dargestellt. Die Streuung für die Zielfunktion 5.2 ist für jedes Modell größer oder nahezu gleich der Streuung von Zielfunktion 5.3.

In den Abbildungen 6.4, 6.5 und 6.6 sind die exemplarisch partitionierten Graphen der drei Modelle dargestellt. Wegen der großen Streuung der Ergebnisse können sich in einigen Fällen auch bei der Durchführung zahlreicher Durchläufe der Partitionierung (hier 30) recht schlechte Ergebnisse als Optimum der Durchläufe ergeben. Wie in Abbildung 6.7 dargestellt, sind dann Partitionen weder logisch noch geometrisch zusammenhängend (in Abbildung mit X markiert).

6.3.2 Bewertung der Ergebnisse

Die KL-Zielfunktion 5.2 erzielt größere Ratio-Cuts als die auf Balance basierende Zielfunktion 5.3 (s. Tabelle 6.2), aber die Zielfunktion 5.3 weist eine kleinere Ergebnisstreuung auf (s. Tabelle 6.3). Da die Streuung nur als sekundäres Bewertungskriterium verwendet wer-

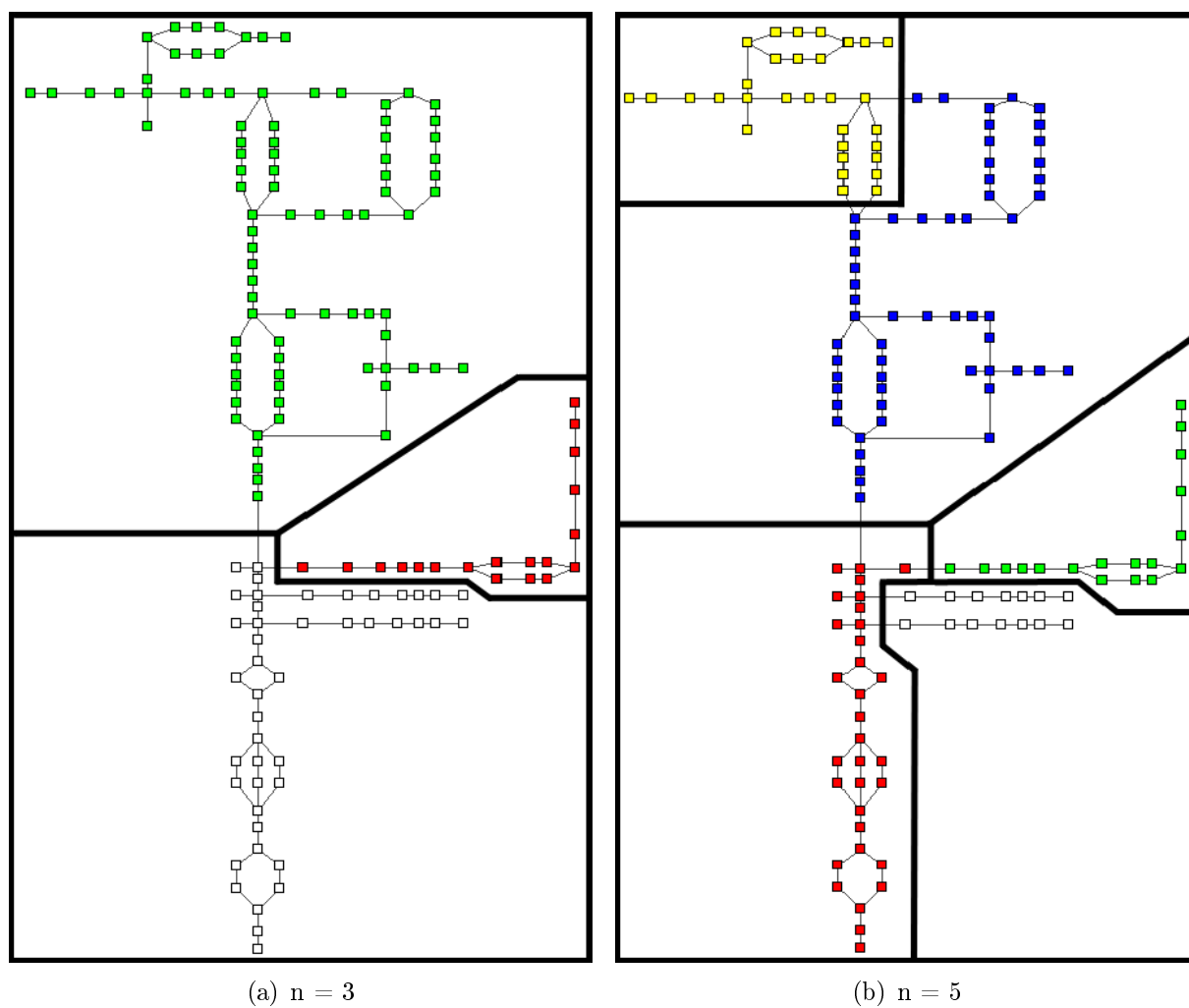


Abbildung 6.4: Partitionierungen Modell A

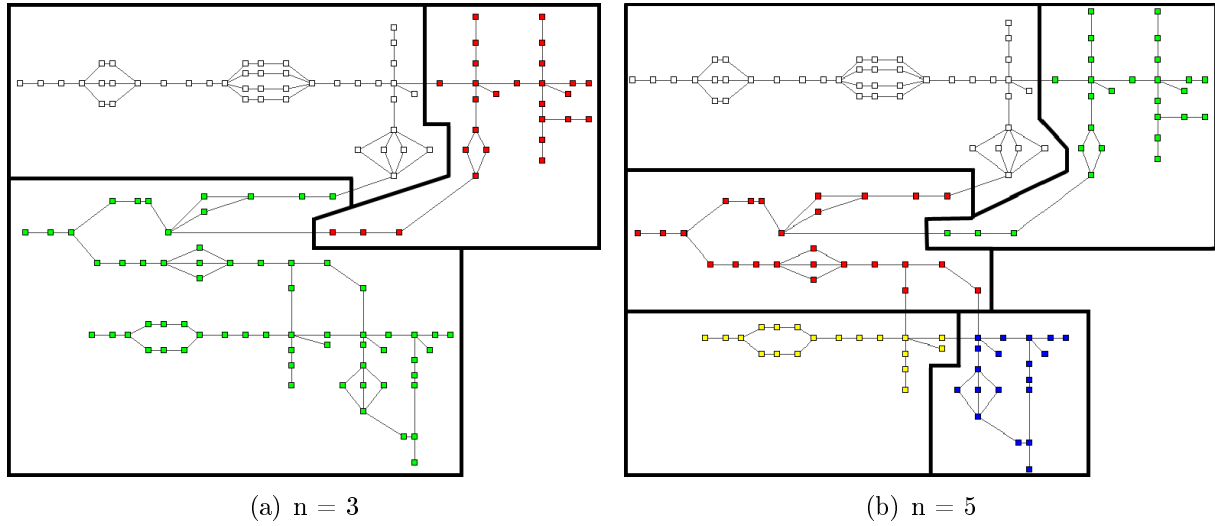


Abbildung 6.5: Partitionierungen Modell B

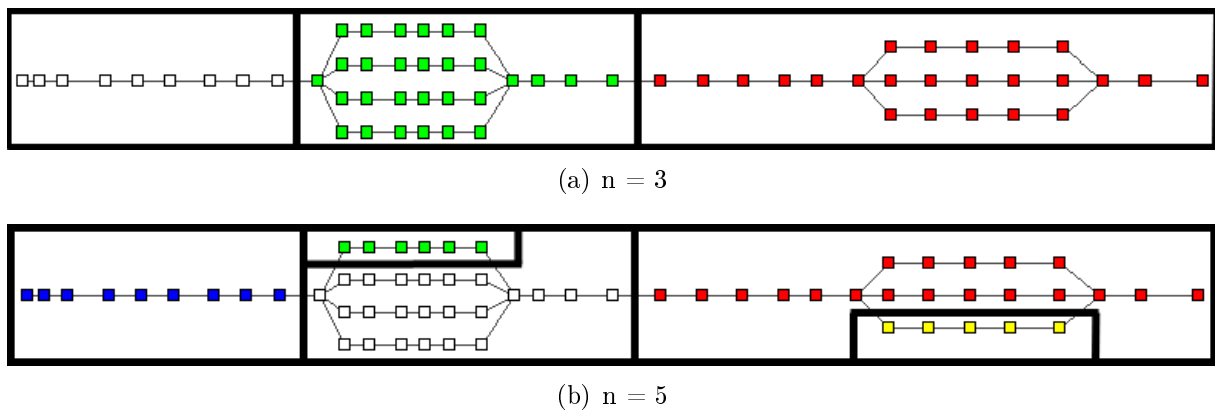


Abbildung 6.6: Partitionierungen Modell C

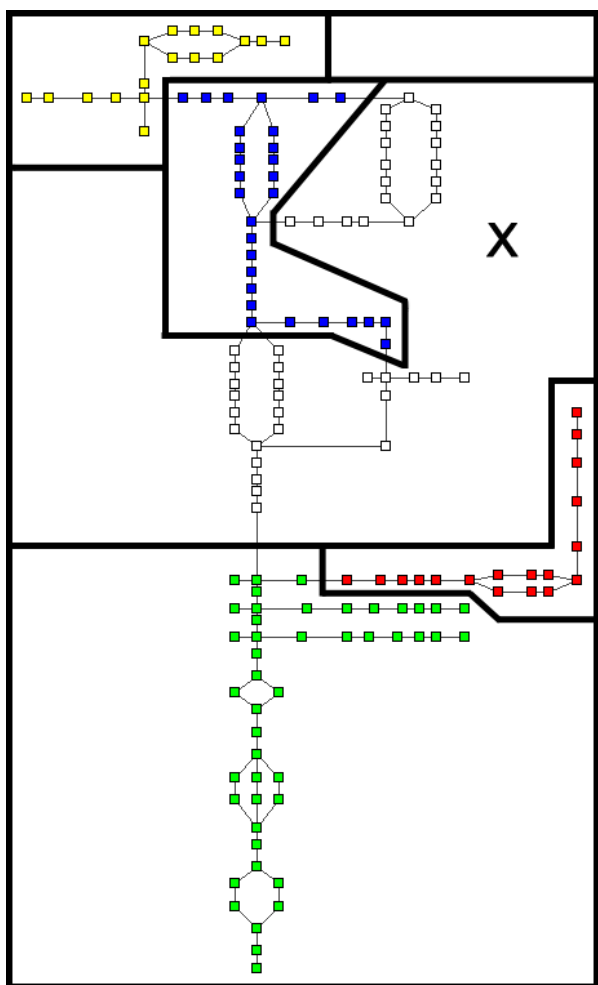
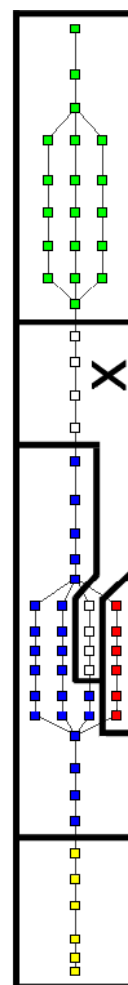
(a) Modell A, $n = 5$ (b) Modell C, $n = 5$

Abbildung 6.7: Partitionierungen mit schlechtem Ergebnis

den sollte, ist die KL-Zielfunktion 5.2 die bessere der beiden. Um die große Streuung zu berücksichtigen und schlechte Ergebnisse zu vermeiden (vgl. Abbildung 6.7), sollten zur Partitionierung sehr viele Durchläufe (> 100) durchgeführt werden. Da die Laufzeit des Partitionierungsalgorithmus aufgrund der Verwendung eines Multilevel-Verfahrens sehr gut ist, stellt diese Bedingung keine besondere Schwierigkeit dar.

Die in den Abbildungen 6.4, 6.5 und 6.6 dargestellten Partitionierungen erfüllen die Anforderungen 6, 7 und 8. Deutlich wird aber, dass die Balance mit steigender Partitionsanzahl schlechter wird, der geometrische und logistische Zusammenhang jedoch immer gegeben ist.

6.4 Validierung der Vereinfachungsmethode

Zur Validierung der Anforderungen 11 und 12 sollen Versuchsreihen mit allen drei Ausgangsmodellen durchgeführt werden. Die Ausgangsmodelle sollen mehrstufig vereinfacht werden. Konnten viele unterschiedliche Vereinfachungen aus einem Ausgangsmodell erzeugt werden, so soll Anforderung 11 als erfüllt gelten. Darauf folgend sollen ausgewählte Vereinfachungen und das Ausgangsmodell mehrfach¹ simuliert und deren Verhalten aufgezeichnet werden. Um zu überprüfen, ob Anforderung 12 erfüllt wird, sollen die Trajektorien der Kennzahlen CT, WIP und TP des Ausgangsmodells mit denen der Vereinfachungen verglichen werden. Ist die Schwankungsbreite der Trajektorien der Vereinfachungen ähnlich denen des Ausgangsmodells und nimmt diese nicht ab, so soll die Anforderung als erfüllt gelten. Zusätzlich soll noch überprüft werden, ob die Engstellenberechnung nach Formel 5.5, die eine zentrale Bedeutung in der Vereinfachung besitzt, die Engstellen im Modell zuverlässig bestimmt. Dazu sollen die berechneten Werte von $C_{eff,rel}$ mit simulativ bestimmten Auslastungen von Maschinen verglichen werden.

6.4.1 Ergebnisse

Die Komplexität von Teilmodellen im Verlaufe der Vereinfachung für alle drei Modelle wird in Abbildung 6.8 dargestellt. Zur Aufzeichnung wurden in jeder Vereinfachungsste-

¹Jedes Modell wurde drei Mal simuliert, wegen der Zufallsabhängigkeit der Simulation unter Verwendung unterschiedlicher Zufallszahlenströme.

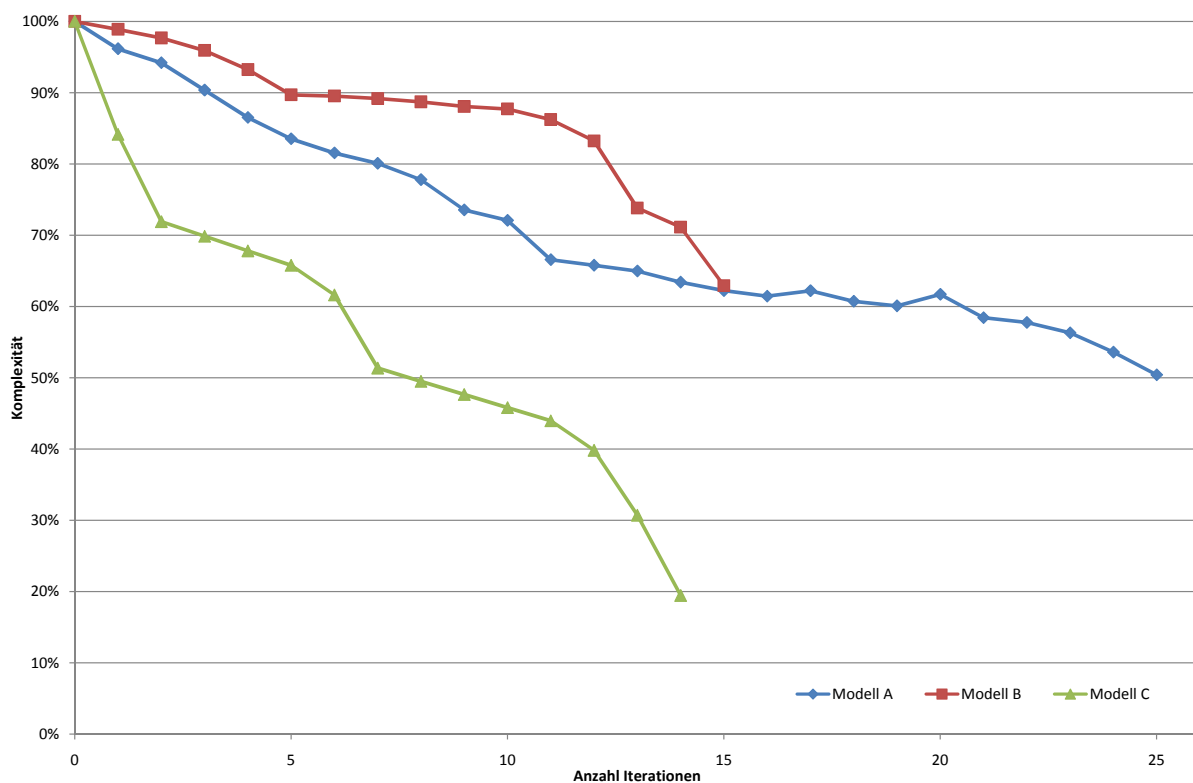


Abbildung 6.8: Komplexität im Verlauf der Vereinfachung

ration 2-6 Vereinfachungsoperationen zugelassen und nach jeder Iteration die Komplexität gemessen.

Es wurden drei Vereinfachungen unterschiedlicher Komplexität ausgewählt, simuliert und das Verhalten gemessen. Aus den aufgezeichneten Kennzahlen wurden die Variationskoeffizienten dieser bestimmt, welche in Tabelle 6.4 dargestellt sind. Die Variationskoeffizienten des Durchsatzes sind in allen Modellen identisch, dies ist dadurch begründet, dass der Durchsatz eine stetig steigende Funktion ist, bei der die Schwankung um den Mittelwert keine Aussage liefert. Der Durchsatz kann somit zur Validierung der Anforderung nicht verwendet werden. Bei den beiden anderen Kennzahlen steigt der Variationskoeffizient mit steigender Vereinfachung im Allgemeinen leicht. Die Schwankungsbreite der Kennzahlen nimmt also leicht zu. Der mittlere Fehler zwischen Variationskoeffizient des Ausgangsmodells und der Vereinfachungen beträgt 1,46%².

Zur Bestimmung der Güte von Formel 5.5 wurde in allen Ausgangsmodellen die Auslastung der Maschinen simulativ bestimmt, in einer Liste eingetragen und diese der Größe

²Ohne Verwendung der Kennzahl TP

Tabelle 6.4: Erhalt der Stochastik

Modell	Kennzahl	Variationskoeffizient				mittl. Fehler
		Ω	b_2	b_1	b_0	
A	CT	0,177	0,180	0,124	0,201	2,33%
A	WIP	0,226	0,216	0,234	0,238	0,75%
A	TP	0,577	0,577	0,577	0,577	0,00%
B	CT	0,477	0,498	0,483	0,488	0,65%
B	WIP	0,242	0,247	0,313	0,322	3,65%
B	TP	0,577	0,577	0,577	0,577	0,00%
C	CT	0,181	0,174	0,155	0,200	1,3%
C	WIP	0,145	0,148	0,146	0,146	0,09%
C	TP	0,577	0,577	0,577	0,577	0,00%

Tabelle 6.5: Güte der Engstellenberechnungsfunktion

Modell	Korrelationskoeffizient
A	0,63
B	0,04
C	1,00

nach sortiert. Jede Maschine erhielt daraufhin eine Positionsnummer, die mit dem größten Wert erhielt die Nummer 1, die zweitgrößte die 2 usw. Analog dazu wurde mit den für die Maschinen berechneten $C_{eff,rel}$ verfahren. Dann wurde jeder Maschine in der Liste der Auslastungen die Position in der Liste der $C_{eff,rel}$ zugewiesen und für diese Zuordnung wurde der Korrelationskoeffizient bestimmt. Würde die Berechnung die Auslastungen perfekt vorhersagen, müsste sich ein Korrelationskoeffizient von ≈ 1 ergeben. Es ergaben sich die Werte in Tabelle 6.5. Die Engstellen werden in Modell C perfekt vorhergesagt und die Bestimmung in Modell B entspricht in der Güte einer Zufallsauswahl. Modell A liegt zwischen diesen beiden Fällen.

6.4.2 Bewertung der Ergebnisse

Zur Aufzeichnung von Abbildung 6.8 erfolgten Vereinfachungen in 14 oder mehr Stufen. Somit könnten deutlich mehr unterschiedliche Teilmodelle erzeugt werden als in einer praktischen Anwendung der dvdS sinnvoll wären, deshalb soll dies als Beweis der nahezu vollständigen Stufenlosigkeit gelten und Anforderung 11 erfüllt sein. Die Variationskoeffizienten der Kennzahlen steigen mit zunehmender Vereinfachung sehr leicht an, warum dies der Fall ist, kann nicht geklärt werden u.U. handelt es sich bei der kleinen Zahl an

Messungen auch nur um zufällige Abweichungen. Der durchschnittliche Fehler der Variationskoeffizienten der Vereinfachungen im Verhältnis zum Ausgangsmodell ist mit 1,46% so gering, dass von einer gleichbleibenden Zufallsabhängigkeit der Kennzahlen ausgegangen werden kann und Anforderung 12 erfüllt ist.

Die Messung der Engstellenbestimmungsgüte durch Formel 5.5 liefert keine eindeutigen Ergebnisse, dafür müssten mehr Modelle untersucht werden. Mit mehr Versuchen könnte eine Aussage darüber getroffen werden, ob die Güte der Engstellenbestimmung abhängig ist vom durchschnittlichen Knotengrad (der Verzweigtheit) der Komponenten im Modell. Im Modell C ist der durchschnittliche Knotengrad wesentlich kleiner als bei Modell A und dort wiederum kleiner als in Modell B, weil verhältnismäßig weniger Verteiler, Zusammenführer und Montagemaschinen enthalten sind. Bei den untersuchten drei Modellen nimmt die Güte der Engstellenbestimmung mit Zunahme des durchschnittlichen Knotengrades ab.

6.5 Validierung der Regelung

Zur Validierung der Anforderungen 13 und 14 sollen alle drei Ausgangsmodelle mehrstufig vereinfacht werden. Für alle erzeugten Vereinfachungen soll die Verhaltensabweichung mit dem in Abschnitt 5.5.3 dargestellten Verfahren in mehreren Simulationsläufen gemessen werden. Aus den Ergebnissen soll dann ein Diagramm analog zu Abbildung 5.7 erzeugt werden. Ergibt sich ein ähnlicher Verlauf der Komplexität über der Verhaltensabweichung, dann soll Anforderung 13 als erfüllt gelten, weil Kombinationen von Zielen für Komplexität und Verhaltensabweichung wie konzipiert erreicht werden können. Weil bei Johnson et al. [JFM05] die Verhaltensabweichung von der Auslastung des Produktionssystems abhängig ist, sollen zusätzlich noch Versuche durchgeführt werden, um einen solchen Zusammenhang zu finden. Für alle erzeugten Modelle und die Ausgangsmodelle soll die Komplexität berechnet und die Laufzeit auf einem einzigen Rechner gemessen werden. Wegen zufälliger Einflüsse auf die Ausführungsgeschwindigkeit sollen auch diese Messungen mehrfach erfolgen. Die Anforderung 14 soll als erfüllt gelten, wenn sich ein Korrelationskoeffizient zwischen Komplexität und Laufzeit größer als 0,8³ ergibt.

³Soll als erste Näherung für starke Korrelation gelten, ohne Berücksichtigung weiterer statistischer Nebenbedingungen und Tests.

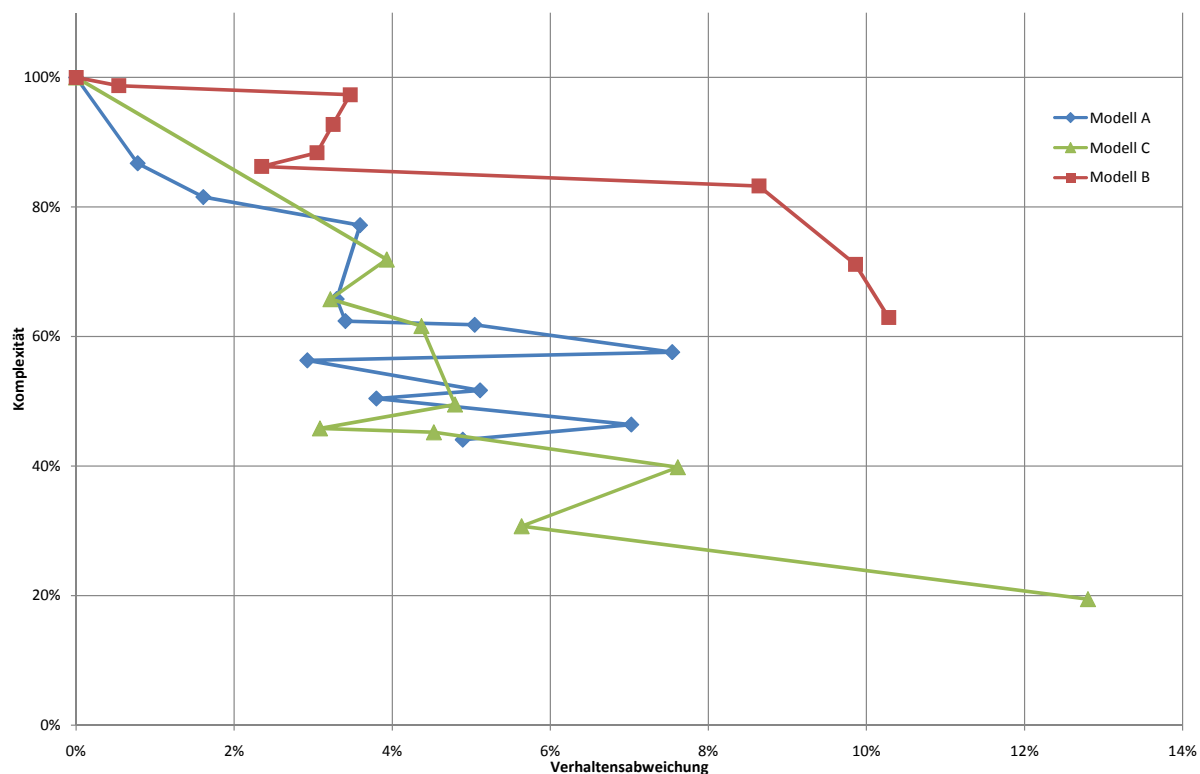


Abbildung 6.9: Komplexität über Verhaltensabweichung von vereinfachten Modellen

6.5.1 Ergebnisse

Die Messung der Verhaltensabweichungen und Komplexität wird in Abbildung 6.9 dargestellt.

Die Laufzeitmessungen wurden auf einem Rechner mit einem Intel E8100 (2,1 GHz) Prozessor und 2,8 GB RAM unter Fedora Linux 9 durchgeführt. Es ergaben sich für die drei Modelle die Werte in den Tabellen 6.6, 6.7 und 6.8.

Für Modell A ergibt sich ein Korrelationskoeffizient von 0,99, für Modell B von 0,83 und für Modell C von 0,98. Die Korrelation von Laufzeit und Komplexität wird beispielhaft für Modell A in Abbildung 6.10 verdeutlicht.

Zur Untersuchung des Einflusses der Auslastung des Produktionssystems auf die Verhaltensabweichung wurde das Zeitintervall zwischen zwei Marken in den Quellen um 20% gesenkt und erhöht. Wird das Zeitintervall erhöht, sinkt die Auslastung des Systems, im anderen Fall erhöht sie sich. Die Ausgangskonfiguration der Modelle erfolgte allerdings

Tabelle 6.6: Korrelationsmessung zwischen Maß und Laufzeit Modell A

Komplexitätsstufe	Komplexitätsmaß	Laufzeit in s			
		M1	M2	M3	Ø
11	317,6	135,0	131,0	130,0	132,0
10	275,5	113	106,0	106,0	108,3
9	259,0	107,1	103,3	101,9	104,1
8	245,2	97	95,1	100,1	97,4
7	208,9	77,1	77,4	78,0	77,5
6	200,1	76,3	75,8	75,8	76,0
5	197,6	80,2	78,1	79,3	79,2
4	186,6	78,6	75,5	74,5	76,2
3	181,6	73,3	70,3	68,7	70,8
2	166,9	64,3	64,3	64,9	64,5
1	162,9	67,2	65,1	62,6	65,0

Tabelle 6.7: Korrelationsmessung zwischen Maß und Laufzeit Modell B

Komplexitätsstufe	Komplexitätsmaß	Laufzeit in s			
		M1	M2	M3	Ø
7	171,8	218,3	214,3	215,6	216,1
6	170,0	193,4	191,4	192,3	192,4
5	163,9	186,7	185,7	187,1	186,5
4	151,2	171,3	171,5	171,3	171,4
3	145,7	164,6	166,6	165,5	165,4
2	122,1	160,5	161,0	160,8	160,8
1	108,1	161,2	160,6	160,2	160,7

Tabelle 6.8: Korrelationsmessung zwischen Maß und Laufzeit Modell C

Komplexitätsstufe	Komplexitätsmaß	Laufzeit in s			
		M1	M2	M3	Ø
7	111,0	66,3	61,9	62,9	63,7
6	77,5	54,2	53,2	51,3	52,9
5	68,9	46,1	45,7	45,9	45,9
4	64,3	42,4	44,2	42,7	43,1
3	48,8	32,1	33,7	33,5	33,1
2	44,2	28,7	27,7	27,8	28,1
1	34,1	21,5	22,1	21,5	21,7

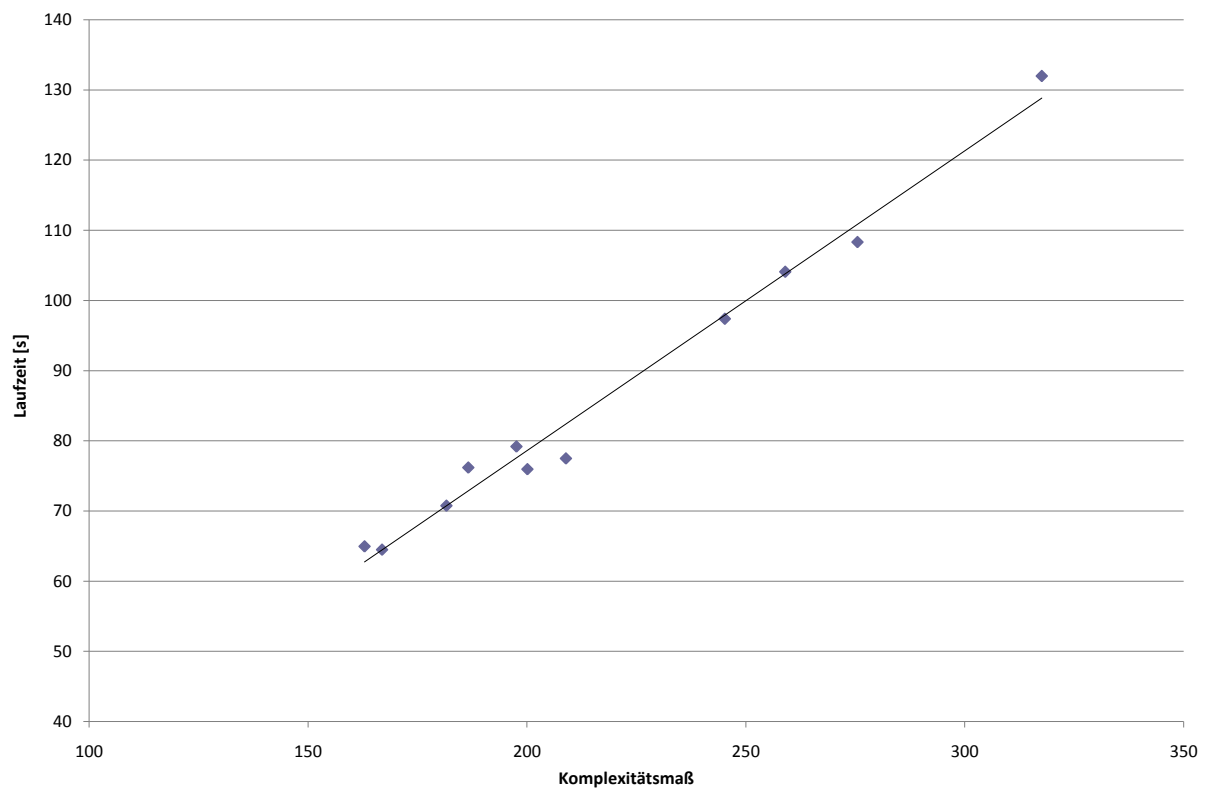


Abbildung 6.10: Komplexitätsmaß und Laufzeit von Modell A

Tabelle 6.9: Abhängigkeit der Verhaltensabweichung von der Auslastung

Modell	mittlerer Fehler in %		
	-20% Zeitintervall	normal	+ 20% Zeitintervall
A	3,4	3,4	2,0
B	2,6	2,6	2,5
C	4,6	5,1	3,4

schon so, dass die Engpässe eine Auslastung von $>90\%$ hatten. Die Auslastung kann also nicht wesentlich gesteigert werden. Es ergaben sich bei Vereinfachungen mittlerer Stärke die Fehler in Tabelle 6.9. Es zeigt sich, dass der Fehler bei einem weniger stark ausgelasteten System deutlich geringer ist.

6.5.2 Bewertung der Ergebnisse

Die Messkurven in Abbildung 6.9 sind stark modellabhängig. Sie bestätigen aber, was in Abschnitt 6.2 angenommen wurde, nämlich dass Modell C stark vereinfacht werden kann, Modell A mittelmäßig und Modell B schlecht. Die Messkurve von Modell C entspricht am deutlichsten dem angenommenen Ideal aus Abbildung 5.7, die Vereinfachung von Modell B zeigt einen ähnlichen Verlauf, aber auf höherem Komplexitätsniveau. Die maximale Vereinfachung von Modell A hat im Vergleich zu den anderen Modellen eine kleine Verhaltensabweichung. Eine weitere Vereinfachung mit u.U. höherer Verhaltensabweichung konnte nicht erzielt werden, weil keine Zusammenfassungen mehr möglich waren und für weitere Weglassungen q_{Grenz} soweit angehoben werden musste, dass das resultierende Modell keine vergleichbaren Simulationsergebnisse mehr erzielen konnte. Anforderung 13 ist somit erfüllt.

Modell A und C haben bei einer Restkomplexität von 45% eine Verhaltensabweichung von ungefähr 5% und die maximal erreichte Vereinfachung besitzt 19% Restkomplexität bei Modell C bei einer Verhaltensabweichung von 13%. Im Vergleich zu den Ergebnissen von Hung und Leachman 3.3.3 erscheinen diese Ergebnisse nicht besonders gut. Bei der festgestellten starken Modellabhängigkeit der Vereinfachung und ohne genaue Kenntnisse des Modells von Hung und Leachman ist dieser Vergleich aber nur bedingt aussagekräftig. Das Ausgangsmodell C umfasst 62 Komponenten (s. Abbildung 6.3) und die maximale Vereinfachung 7 (s. Abbildung 6.11). Die sieben Komponenten setzen sich zusammen aus einer Quelle, einer Senke und einer „Minimallinie“ mit zwei Sequenzern, zwei Förder-



Abbildung 6.11: Maximale Vereinfachung Modell C

bändern und einer Maschine. Weitere Komponenten können nicht weggelassen werden, weil sonst das Modell, wenn es als Teilmodell mit nachfolgenden Teilmodellen betrachtet wird, nicht mehr valide ist. Damit ist die Anzahl Komponenten um 89% gesunken, stärker als die von Hung und Leachman ($1 - (5/30) = 83,3\%$). Warum die Komplexität (anhand der Laufzeit) bei Hung und Leachman allerdings um 91% sinken konnte und hier nur um 80%, ist mit großer Wahrscheinlichkeit bedingt durch die unterschiedlichen Komponenten(-klassen), die verwendet wurden.

Die Korrelationskoeffizienten von Laufzeit und Komplexität sind bei allen Modellen größer als die geforderten 0,8, somit gilt Anforderung 14 als erfüllt. Die Erkenntnis von Johnson et al. [JFM05] konnte nicht bestätigt werden, nach der die Verhaltensabweichung mit steigender Auslastung geringer wird. Das genaue Gegenteil wurde festgestellt. Da alle hier verwendeten Modelle standardmäßig fast mit maximaler Auslastung konfiguriert sind, sind alle festgestellten Verhaltensabweichungen als schlechteste mögliche Werte anzusehen.

6.6 Validierung der Zustandsgenerierungsmethode

Zur Validierung der Anforderung 16 sollen von jedem der drei Ausgangsmodelle zwei Vereinfachungen erzeugt werden, eine schwache (b_1) und eine starke Vereinfachung (b_2). Die Validierung soll in Anlehnung an die Konsistenzanforderungen von Davis und Palmore erfolgen, wie sie in Abschnitt 3.5.2 dargestellt sind. Dabei sollen die Zustände verglichen werden, die entstehen, wenn die Pfade 2 und 4 der Tabelle verfolgt werden. Dazu wird für den zweiten Pfad das Ausgangsmodell von t_0 bis zu einem bestimmten Zeitpunkt t_1 simuliert, dann findet die Umschaltung auf eine Vereinfachung statt. Diese wird dann bis zum Zeitpunkt t_2 simuliert, an dem dann wieder auf das Ausgangsmodell umgeschaltet wird. Hier wird dann der Zustand gespeichert und mit dem Zustand verglichen, den das Ausgangsmodell annimmt, wenn es ohne Umschalten von t_0 bis t_2 simuliert wird. Für den

Tabelle 6.10: Abweichung des Durchsatzes nach Umschalten von Teilmodellen

Modell	rel. Fehler [%]			
	Pfad 2		Pfad 4	
	b_1	b_2	b_1	b_2
A	0,70	4,70	1,00	6,11
B	1,84	3,04	6,16	0,19
C	3,48	3,51	3,03	2,61

vierten Pfad erfolgt ein analoges Vorgehen. Zum Vergleichen der Zustände werden die Werte der Statistikvariablen aller Komponenten des Modells verwendet. Dieses Vorgehen zur Validierung wurde gewählt, weil es ein bekanntes Verfahren ist und es erlaubt, die Folgen der Umschaltung durch den Vergleich zweier Zustände desselben Modells zu erfassen. Da das aktivierte und das deaktivierte Modell nur in wenigen Bereichen identisch sind, kann ein direkter Vergleich der Zustände vor und nach der Umschaltung nicht erfolgen. Beim Vergleich der Zustände zum Zeitpunkt t_2 wird der Unterschied der Zustände beeinflusst durch die Zustandsabbildung und durch die Simulation im Zeitraum $t_2 - t_1$. Um den Fehler der reinen Zustandsabbildung zu untersuchen, sollen zusätzlich Versuche durchgeführt werden, bei dem $t_2 - t_1 = 0$ ist.

6.6.1 Ergebnisse

Die Zustände zum Zeitpunkt t_2 wurden verglichen und die in Tabelle 6.10 dargestellten relativen Fehler gemessen. Bei den durchgeführten Versuchen wurden t_1 und t_2 so gewählt, dass gilt $t_1 - t_0 = t_2 - t_1$. Wie in Abschnitt 5.6 dargestellt, wird der Durchsatz als einzige Statistikvariable uneingeschränkt vom deaktivierten auf das aktivierte Teilmodell übertragen. Die übrigen Statistikvariablen werden nur bei in beiden Modellen „unverändert“ vorhandenen Komponenten übertragen, diese zeigen aber einen vergleichbaren relativen Fehler wie der Durchsatz.

Aus Tabelle 6.10 wird ersichtlich, dass die Zustandsabbildung funktioniert, also ein valider Zustand im aktivierten Modell erzeugt wird, weil die aktivierten Modelle nach der Umschaltung weiterlaufen. Dies ist hauptsächlich verursacht durch die richtige Platzierung von Marken und die richtige Einstellung der belegungsbedingten Zustände der Komponenten. Nachdem die Simulation die Hälfte der Simulationszeit auf einer Komplexitätsstufe durchgeführt wurde und die andere Hälfte auf der anderen, ergeben sich Fehler im Durchsatz von 0,2% bis 6,2%, der Mittelwert liegt bei 3,03%. Dieser Fehler kann aber nicht

Tabelle 6.11: Fehler beim Umschalten von Teilmodellen ohne zwischenzeitliche Simulation

Modell	relativer Fehler [%]			
	b_1		b_2	
	Durchsatz	Markenbelegung	Durchsatz	Markenbelegung
A	0,04	16,0	1,3	40,0
B	0,03	4,7	2,7	24,8
C	0,25	28,0	1,6	49,0

vollständig der Zustandsabbildung angelastet werden, weil die verwendeten Vereinfachungen eine Verhaltensabweichung zum Ausgangsmodell aufweisen, die bei den verwendeten Vereinfachungen ebenfalls im Bereich um 4% liegt.

Gilt $t_2 - t_1 = 0$ und erfolgt das Umschalten analog zu Pfad 1, ergeben sich die in Tabelle 6.11 dargestellten Fehler, wobei der Fehler der Markenbelegung nicht die Anzahl Marken betrifft, diese bleibt immer gleich, sondern die Komponenten, die sie belegen. Der Fehler gibt an, wie viele Marken nach den Umschaltungen nicht mehr auf denselben Komponenten liegen wie vorher. Der Fehler des reinen Umschaltens liegt somit bei 2% bis 89% des Gesamtfehlers aus Tabelle 6.10.

6.6.2 Bewertung der Ergebnisse

Die entwickelte Zustandsabbildungsfunktion erzeugt im zu aktivierenden Teilmodell einen validen Zustand, das bedeutet, dass Marken, belegungsbedingte Zustände und Ereignisse korrekt übertragen werden. Nach zweifacher Umschaltung und Simulation des aktivierten Modells ergeben sich Fehler gegenüber einer Simulation ohne Umschalten von 0,2% bis 6,2%. Die Streuung der Werte ist sehr hoch und die Anzahl Messwerte klein, daher ist der Median von 3,0% mit Vorsicht zu genießen. Dieser Fehler liegt im Bereich, der bei der Vereinfachung von Modellen als akzeptabel eingestuft wurde. Zusätzlich wird dieser Fehler durch die Simulation mit verursacht und nicht nur allein durch das Umschalten. Der Fehler des Umschaltens liegt zwischen 0,03% und 2,7%, dies macht 2% bis 89% des Gesamtfehlers aus. Im Gegensatz zu Tabelle 6.10 (hier gilt es nur für 66%) ist in Tabelle 6.11 die klare Tendenz zu erkennen, dass die Fehler beim Umschalten stark vereinfachter Modelle größer sind als bei weniger stark vereinfachten Modellen. Dies liegt an der geringeren Anzahl unverändert im aktivierten und deaktivierten Modell vorhandenen Komponenten. Die Zustandsgenerierung soll als valide gelten und Anforderung 16 erfüllt sein.

7 Zusammenfassung und Ausblick

First come smiles,
then lies.
Last is gunfire.

*(Wolves of the Calla,
Stephen King)*

Modelle der Materialflusssimulation werden immer komplexer, weil immer mehr Unternehmensbereiche mittels Simulation untersucht werden sollen und der Anspruch an die Abbildungsgenauigkeit wächst. Mit diesen Modellen lässt sich nicht effizient experimentieren, weil die Laufzeit der Simulation dieser komplexen Modelle sehr groß ist. Eine Möglichkeit, die Laufzeit der Simulation zu verkleinern, ist die Komplexität der vorhandenen Modelle mittels Vereinfachung zu reduzieren. Ein vereinfachtes Modell zeigt aber im Allgemeinen eine Abweichung im Verhalten im Vergleich zum Ausgangsmodell. Dies widerspricht dem Wunsch nach hoher Abbildungsgenauigkeit. Mueck [Mue05] entwickelte ein System, die detaillierungsvariante diskrete Simulation, welche die Laufzeit komplexer Simulationsmodelle diesen Zielen anpasst. Das Problem wird gelöst, indem Teilmodelle geringerer Komplexität verwendet werden, wenn diese nicht im Fokus der Analyse stehen oder keine starken Auswirkungen auf die aktuelle Analyse haben. Eine weitere Eigenschaft des Systems ist, dass der Analysefokus in der detaillierungsvarianten diskreten Simulation verschoben werden kann und dann während der Simulation Teilmodelle höherer Komplexität gegen solche mit niedrigerer Komplexität ausgetauscht werden und andersherum. Modellvereinfachung ist üblicherweise eine manuelle Tätigkeit für Simulationsexperten, die zusätzlich zum Modellerstellungsaufwand anfällt und für das System von Mueck zusätzlich in mehreren Abstufungen erfolgen muss.

7.1 Ergebnis

In dieser Arbeit wurde ein System zur automatischen Erstellung von Modellen zur detaillierungsvarianten diskreten Simulation entwickelt, bei dem der Modellierer nur das Modell mit höchster Komplexität erstellt und vereinfachte Teilmodelle automatisch erzeugt werden. Das Ausgangsmodell wird dazu zuerst iterativ partitioniert, so dass sich eine Hierarchie von immer umfassenderen Partitionen bildet. Dann werden für jede Partition verhaltensähnliche Teilmodelle geringerer Komplexität generiert. Es entsteht eine Hierarchie bezüglich der Komplexität; auf jeder höheren Hierarchiestufe nehmen die Komplexität und die Verhaltensähnlichkeit des Gesamtmodells ab. Um Teilmodelle mit definierten Eigenschaften zu erzeugen, werden die Komplexität und die Verhaltensabweichung als Regelgrößen der Vereinfachung verwendet. Um einen Austausch von Teilmodellen während der Laufzeit der Simulation durchführen zu können, werden Abbildungsfunktionen in die Teilmodelle integriert, die aus dem Zustand der entnommenen Teilmodelle einen Zustand für die eingefügten Teilmodelle berechnen, der dem Zustand der entnommenen Teilmodelle ähnlich ist.

Es wurden fünf Teilprobleme definiert, die gelöst werden mussten, um das oben beschriebene System zu realisieren. Die Teilprobleme sind: Die Modellbeschreibungsmethode, die Partitionierung, die automatische Vereinfachung, die Regelungstechnik und die Zustandsabbildung. Im Folgenden sollen diese Teilprobleme und ihre Lösung kurz beschrieben werden.

Um eine automatische Modellvereinfachung zu erlauben, wurden 18 Komponentenklassen erstellt, aus deren verknüpften und parametrisierten Instanzen ein Modell aufgebaut wird. Diese Modellkomponenten erlauben eine flexible Modellierung unterschiedlicher und praxisrelevanter Fertigungssysteme. Damit ein so modelliertes vereinfachungsfähiges Modell keinen höheren Berechnungsaufwand aufweist als ein nicht vereinfachungsfähiges und damit die Sinnhaftigkeit der Vereinfachung nicht gegeben wäre, wurden in die entwickelten Komponentenklassen keine Daten oder Methoden integriert, die nur für die Vereinfachung genutzt werden.

Zur iterativen Partitionierung wurde ein Multilevelalgorithmus gewählt, der durch eine Balancerestriktion und Minimierung der Schnittgröße unter Verwendung einer längenabhängigen Kantengewichtung ein Modell so partitioniert, dass relativ gleich umfangreiche und logisch und geometrisch zusammenhängende Partitionen gebildet werden. Der Algo-

rithmus liefert ordentliche und den Anforderungen genügende Ergebnisse. Eine Weiterentwicklung, mit besonderer Berücksichtigung des Ziels, optimierte Partitionen für eine Vereinfachung zu bilden, kann die Ergebnisse verbessern. Da die Partitionierung in dieser Arbeit ein vom Rest thematisch entkoppeltes Gebiet darstellt, soll diese Weiterentwicklung durch Spezialisten im Bereich Graphpartitionierung durchgeführt werden.

Zur automatischen und regelbaren Vereinfachung wurden Methoden für die Techniken Zusammenfassung und Weglassung entwickelt. Diese Methoden erfordern keine vor der Vereinfachung stattfindenden Simulationen zur Datenbeschaffung, sondern nutzen nur den Aufbau und die Parametrierung des Ausgangsmodells. Dadurch können leichte bis mittelschwere Änderungen im Ausgangsmodell sehr schnell auf die Vereinfachungen abgebildet werden, dies führt aber zu der Schwierigkeit, die Vereinfachung so zu gestalten, dass sich in der Simulation ähnliche Wartezeiten von Marken auf Komponenten einstellen. Deshalb werden Engstellen im Modell besonders betrachtet und es wurde eine Methode zur Engstellenberechnung weiterentwickelt. Diese identifiziert Komponenten, die in der Vereinfachung bestehen bleiben müssen. Bei der Vereinfachung durch Zusammenfassung werden Strukturen mit vielen Komponenten durch Strukturen mit weniger Komponenten unter der Randbedingung substituiert, dass das Verhalten der beiden Strukturen ähnlich ist. Diese Zusammenfassung erfolgt anhand eines entwickelten Regelsatzes, der für eine Vielzahl von Komponentenkombinationen in zwei Grundstrukturen festlegt, wie die Struktur, die Komponenten und Komponentenparametrierung der Vereinfachung zu gestalten sind. Bei der Vereinfachung durch Weglassung werden Komponenten aus dem Modell gestrichen, wenn ein entwickeltes Maß sie als das Verhalten des Modells nicht wesentlich beeinflussend identifiziert.

Zur Regelung der Vereinfachung wurde ein Regelungsalgorithmus entwickelt, der zur Messung der Komplexität und der Verhaltensabweichung von vereinfachten Modellen zwei Maße benutzt, um den Vorgaben entsprechende Modelle zu erzeugen. Das entwickelte Komplexitätsmaß betrachtet jede Komponente im Modell anhand ihrer Klasse und ihrer Verknüpfung im Modell und berechnet so die Komplexität des Modells und erreicht im Vergleich von Ausgangsmodell zu Vereinfachung eine sehr gute Korrelation zur Laufzeit. Die Messung der Verhaltensabweichung erfolgt durch die Differenzbildung der Integrale der über die Zeit aufgetragenen Kennzahlen Durchsatz, Umlaufbestand und Durchlaufzeit. Der Regelungsalgorithmus umfasst eine feingranulare Vereinfachung in mehreren Iterationen, nach denen jeweils die Komplexität des erzeugten Modells gemessen wird. Dies erfolgt so lange, bis der Zielwert der Komplexität erreicht wird. Im Verlauf der Vereinfachung

kann es erforderlich sein, den Schwellwert des voraussichtlichen Einflusses auf die Verhaltensabweichung der möglichen Vereinfachungsoperationen anzuheben, um eine weitere Vereinfachung zu ermöglichen. Wenn die Komplexität den Zielvorgaben entsprechend gesenkt werden konnte, erfolgt eine simulative Messung der Verhaltensabweichung und damit die Bewertung, ob ein für die detaillierungsvariante diskrete Simulation taugliches Modell erstellt werden konnte. Mit der geregelten Vereinfachung konnten modellabhängig große bis sehr große Komplexitätsreduktionen bei gleichzeitig geringen Verhaltensabweichungen erreicht werden.

Zur Zustandsabbildung werden die Vereinfachungsoperationen benutzt, die zur Erzeugung einer Vereinfachung durchgeführt wurden. Diese definieren eine Erzeugungsbeziehung von Komponenten der Vereinfachung und weisen so Komponentenmengen in der Vereinfachung zugehörige Komponentenmengen im Ausgangsmodell zu. Durch die entwickelte Zustandsabbildungsfunktion kann der Simulationszustand von zugeordneten Komponentenmengen in beide Richtungen, also von komplex zu vereinfacht und andersherum, übertragen werden. Dazu werden zuerst die auf den Komponenten befindlichen Marken, die Statistikvariablen und Störstatus übertragen und dann aus der Markenbelegung weitere Zustandsvariablen und die Liste der eingeplanten Ereignisse berechnet. Die Zustandsabbildung erzeugt konsistente Zustände, so dass die Simulation nach einem Austausch von Teilmodellen weiterläuft, und erzeugt beim Austausch und im weiteren Simulationsverlauf nur geringe Fehler.

7.2 Weiterer Forschungsbedarf

Aus den Ergebnissen dieser Arbeit ergeben sich zwei interessante Aspekte, die weiterer Untersuchung bedürfen. Dies wäre zum einen eine Weiterentwicklung der Modellanalyse im Bereich der automatischen Vereinfachung, um die Verhaltensabweichung durch eine bessere Bewertung von Wartezeiten zu verringern, und zum anderen die Neuentwicklung eines Verfahrens zur Onlinevereinfachung, die auf der automatischen Vereinfachung basiert.

Als Onlinevereinfachung soll eine Vereinfachung verstanden werden, die keine Menge an vereinfachten Modellen vor Beginn der Simulation bereitstellt, sondern erst während der Simulation gestartet wird. Die Onlinevereinfachung verringert die Gesamtlaufzeit des Bündels aus Vereinfachung und detaillierungsvarianter diskreter Simulation und erlaubt eine

viel stärker auf den Analysefokus der Simulation hin ausgerichtete Vereinfachung, die eine größere Komplexitätsreduktion durch größere Partitionen verspricht. Eine Onlinevereinfachung erscheint möglich, da die Methoden zur automatischen Vereinfachung keine Simulationen zur Datengewinnung benötigen und dadurch die Laufzeit der Vereinfachung auch bei großen Modellen gering ist. Im Falle einer Onlinevereinfachung könnte allerdings keine Regelung anhand der Verhaltensabweichung erfolgen, da diese die Durchführung von zusätzlichen Simulationen gebietet. Eine Vorbedingung muss also sein, dass zuverlässig abgeschätzt werden kann, bis zu welcher Komplexität vereinfacht werden kann und welche Vereinfachungsoperationen auf Strukturen angewandt werden können, um eine bestimmte Verhaltensabweichung nicht zu überschreiten. Dazu muss die Modellabhängigkeit der Vereinfachungsfähigkeit umfassend verstanden werden, was besonders über eine bessere Bewertung von Wartezeiten erfolgen muss, da diese sich nicht alleinig aus der Parametrierung der Komponenten, sondern zusätzlich aus deren Verknüpfung und einer wechselseitigen Beeinflussung ergeben.

Literaturverzeichnis

- [AK95] ALPERT, Charles J. ; KAHNG, Andrew B.: Recent Directions in Netlist Partitioning: A Survey. In: : *Integration, the VLSI Journal* 19 (1995), S. 1–81
- [Bar92] BARTON, Russell R.: Metamodels for Simulation Input-Output Relation. In: *Proceedings of the 1992 Winter Simulation Conference*, 1992, S. 289–299
- [Bar98] BARTON, Russel R.: Simulation Metamodels. In: *Proceedings of the 1998 Winter Simulation Conference*, 1998, S. 167–174
- [BB98] BATTITI, R. ; BERTOSSI, A. A.: Differential Greedy for the 0-1 Equicut Problem. In: *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science* 40 (1998), S. 3–22
- [Bir91] BIROLINI, Alessandro: *Qualität und Zuverlässigkeit technischer Systeme*. Springer, 1991
- [BS94] BARNARD, Stephen T. ; SIMON, Horst D.: A Fast Multilevel Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems. In: *Concurrency : Practice and Experience* 6 (1994), S. 101–117
- [BT00] BROOKS, Roger J. ; TOBIAS, Andrew M.: Simplification in the simulation of manufacturing systems. In: *International Journal of Production Research* 38 (2000), Nr. 5, S. 1009–1027
- [Bus95] BUSS, Arnold H.: A Tutorial on Discrete-Event Modeling with Simulation Graphs. In: *Proceedings of the 1995 Winter Simulation Conference*, 1995, S. 74–81
- [Bus96] BUSS, Arnold H.: Modeling with Event Graphs. In: CHARNES, J. M. (Hrsg.) ; MORRICE, D. J. (Hrsg.) ; BRUNNER, D. T. (Hrsg.) ; SWAIN, J. J. (Hrsg.): *Proceedings of the 1996 Winter Simulation Conference*, 1996, S. 153–160
- [CSZ93] CHAN, Pak K. ; SCHLANG, Martine D. F. ; ZIEN, Jason Y.: Spectral K-Way Ratio-Cut Partitioning and Clustering. In: *Proceedings of the 30th International Conference on Design Automation*, 1993, S. 749–754
- [DAL⁺06] DANGELMAIER, Wilhelm ; AUFENANGER, Mark ; LAROQUE, Christoph ; HUBER, Daniel ; MAHAJAN, Kiran: A Multi-User Modeling Environment to Create and Parametrize Material Flow Models in d³FACT insight. In: BAAKE, Uwe (Hrsg.) ; YUCESAN, Enver (Hrsg.): *Proceedings of the 13th European Concurrent Engineering Conference*, 2006, S. 132–137

- [Dav92] DAVIS, P. K.: An Introduction in Variable-Resolution Modeling and Cross-Resolution Model Connection. In: DAVIS, P. K. (Hrsg.) ; HILLESTAD, XXX (Hrsg.): *Proceedings of the Conference on Variable-Resolution Modeling*, 1992
- [DFH⁺08] DANGELMAIER, Wilhelm ; FISCHER, Matthias ; HUBER, Daniel ; LAROQUE, Christoph ; SUESS, Tim: Aggregated 3D-Visualization of a Distributed Simulation Experiment of a Queuing System. In: MASON, S. J. (Hrsg.) ; HILL, R. (Hrsg.) ; MOENCH, L. (Hrsg.) ; ROSE, O. (Hrsg.): *Proceedings of the 2008 Winter Simulation Conference*, 2008, S. 2012–2020
- [DHLM05] DANGELMAIER, Wilhelm ; HUBER, Daniel ; LAROQUE, Christoph ; MUECK, Bengt: d³FACT insight - An immersive material flow simulator with multi-user support. In: BRUZZONE, Agostino (Hrsg.) ; WILLIAMS, Edward (Hrsg.): *Proceedings of the 2005 Summer Computer Simulation Conference*, 2005, S. 239–242
- [DM04] DANGELMAIER, Wilhelm ; MUECK, Bengt: Using dynamic multiresolution modelling to analyse large material flow system. In: INGALLS, R.G. (Hrsg.) ; ROSSETTI, M.D. (Hrsg.) ; SMITH, J.S. (Hrsg.) ; PETERS, B.A. (Hrsg.): *Proceedings of the 2004 Winter Simulation Conference*, 2004, S. 1720–1727
- [DMMH05] DANGELMAIER, Wilhelm ; MAHAJAN, Kiran ; MUECK, Bengt ; HUBER, Daniel: d³FACT insight - Simulation of huge scale models in cooperative work. In: KRUEGER, Joerg (Hrsg.) ; LISOUNKIN, Alexei (Hrsg.) ; SCHRECK, Gerhard (Hrsg.): *EUROSIS-ETI (2005)*, 2005, S. 150–158
- [DMP94] DIEKMANN, Ralf ; MONIEN, Burkhard ; PREIS, Robert: Using Helpful Sets to Improve Graph Bisections / Universität Paderborn. 1994. – Forschungsbericht
- [Dut93] DUTT, Shantanu: New Faster Kernighan-Lin-Type Graph-Partitioning Algorithms. In: *Proceedings of the 1993 IEEE/ACM international conference on Computer-aided design*, IEEE Computer Society Press, 1993, S. 370–377
- [DW97] DANGELMAIER, Wilhelm ; WARNECKE, Hans-Jürgen: *Fertigungslenkung*. Springer, 1997
- [Els97] ELSNER, Ulrich: Graph Partitioning : A Survey / Technische Universität Chemnitz. 1997. – Forschungsbericht
- [FHD⁺00] FISCHER, J. ; HEROLD, W. ; DANGELMAIER, W. ; NASTANSKY, L. ; SUHL, L.: *Bausteine der Wirtschaftsinformatik, 2. überarbeitete und erweiterte Auflage*. Erich Schmidt Verlag, 2000
- [Fis89] FISHWICK, Paul A.: Neural Network Models in Simulation: A Comparison with traditional Modeling Approaches. In: *Proceedings of the 1989 Winter Simulation Conference*, 1989

- [Fjä98] FJÄLLSTRÖM, Per-Olof: Algorithms for Graph Partitioning : A Survey / Linköping Electronic Articles in Computer and Information Science 3. 1998 (10). – Forschungsbericht
- [Foo74] FOO, Norman Yeow-Khean: *Homomorphic Simplification of Systems*, University of Michigan, Diss., 1974
- [Foo01] FOO, Norman Y.: Why Engineering Models do not have a Frame Problem. In: SARJOUGHIAN, Hessam S. (Hrsg.) ; CELLIER, Francois E. (Hrsg.): *Discrete Event Modeling and Simulation Technologies*. Springer, 2001
- [Fra95] FRANTZ, Frederick K.: A Taxonomy of Model Abstraction Techniques. In: *Proceedings of the 1995 Winter Simulation Conference*, 1995, S. 1413–1420
- [Fuj98] FUJIMOTO, Richard M.: Parallel and Distributed Simulation. In: BANKS, Jerry (Hrsg.): *Handbook of Simulation*. John Wiley & Sons, 1998, S. 429–464
- [Gab90] GABOW, Harold N.: Data Structures for Weighted Matching and Nearest Common Ancestors with Linking. In: *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, 1990, S. 434–443
- [GG92] GLASER, Horst ; GEIGER, Volker: *PPS Grundlagen - Konzepte - Anwendungen*. Gabler, 1992
- [GJ79] GAREY, Michael R. ; JOHNSON, David S.: *Computers and Intractability : A Guide to the Theory of NP-Completeness*. Freeman, 1979
- [Gup97] GUPTA, A.: Fast and Effective Algorithms for Graph Partitioning and Sparse-Matrix Ordering. In: *IBM Journal of Research and Development* 41 (1997), S. 171–183
- [HB00] Kap. Metamodellierung zur Analyse des Modellverhaltens In: HUBER, K.-P. ; BERTHOLD, M. R.: *Modellierung, Simulation und Künstliche Intelligenz*. SCS, 2000 (Frontiers in Simulation), S. 243–268
- [HH01] HAHN, Dietger ; HUNGENBERG, Harald: *PuK Wertorientierte Controlling-konzepte*. Gabler, 2001
- [HL99] HUNG, Y.-F. ; LEACHMAN, R. C.: Reduced simulation models of wafer fabrication facilities. In: *International Journal of Production Research* 37 (1999), Nr. 12, S. 2685–2701
- [IMW96] INGALLS, Ricki G. ; MORRICE, Douglas J. ; WHINSTON, Andrew B.: Eliminating Canceling Edges from the Simulation Graph Model Theory. In: *Proceedings of the 1996 Winter Simulation Conference*, 1996, S. 825–832
- [JCS01] JIN, R. ; CHEN, W. ; SIMPSON, T.W.: Comparative studies of metamodelling techniques under multiple modelling criteria. In: *Structural and Multidisciplinary Optimization* 23 (2001), S. 1–13

- [JFM05] JOHNSON, Rachel T. ; FOWLER, John W. ; MACKULAK, Gerald T.: A Discrete Event Simulation Model Simplification Technique. In: *Proceedings of the 2005 Winter Simulation Conference*, 2005, S. 2172–2176
- [JLGL99] JAIN, Sanjay ; LIM, Chu-Chew ; GAN, Boon-Ping ; LOW, Yoke-Hean: Criticality of Detailed Modeling in Semiconductor Supply Chain Simulation. In: *Proceedings of the 1999 Winter Simulation Conference*, 1999, S. 888–896
- [KB87] KLAUS, Georg ; BUHR, Manfred: *Philosophisches Wörterbuch*. Verlag Das Europ. Buch, 1987
- [KG99] KLINGSTAM, P. ; GULLANDER, P.: Overview of simulation tools for computer aided production engineering. In: *Computers in Industry* 38 (1999), S. 173–186
- [Kie97] KIENCKE, Uwe: *Ereignisdiskrete Systeme. Modellierung und Steuerung verteilter Systeme*. Oldenbourg Verlag, 1997
- [KK95] KARYPIS, George ; KUMAR, Vipin: Analysis of Multilevel Graph Partitioning. In: *Proceedings of the 1995 ACM/IEEE conference on Supercomputing*, 1995, S. 96–129
- [KK98a] KARYPIS, George ; KUMAR, Vipin: A fast and high quality multilevel scheme for partitioning irregular graphs. In: *SIAM Journal on Scientific Computing* 20 (1998), S. 359–392
- [KK98b] KARYPIS, George ; KUMAR, Vipin: Multilevel k-way partitioning scheme for irregular graphs. In: *Journal of Parallel and Distributed Computing* 48 (1998), S. 96–129
- [KL70] KERNIGHAN, B. W. ; LIN, S: An Efficient Heuristic Procedure for Partitioning Graphs. In: *The Bell System Technical Journal* 49 (1970), S. 291–307
- [KLCZ92] KIM, T. G. ; LEE, C. ; CHRISTENSEN, E. R. ; ZEIGLER, B. P.: System Entity Structuring and Model Base Management. In: *Proceedings of the Conference on Variable-Resolution Modeling*, 1992, S. 96–101
- [Kle99] KLEIJNEN, Jack P. C.: Validation of Models: Statistical Techniques and Data Availability. In: *Proceedings of the 1999 Winter Simulation Conference*, 1999, S. 647–654
- [KP04] KIESLING, Tobias ; POHL, Siegfried: Time-Parallel Simulation with Approximative State Matching. In: *Proceedings of the 18th Workshop on Parallel and Distributed Simulation*, 2004, S. 195–202
- [Lar07] LAROQUE, Christoph: *Ein mehrbenutzerfähiges Werkzeug zur Modellierung und richtungsoffenen, ereignisorientierten Simulation von wahlweise objekt- und funktionsorientiert gegliederten Fertigungssystemen*, University of Paderborn, Diss., 2007

- [LF96] LEE, Kangsun ; FISHWICK, Paul A.: Dynamic Model Abstraction. In: *Proceedings of the 1996 Winter Simulation Conference*, 1996
- [LK00] LAW, Averill M. ; KELTON, W. D.: *Simulation Modeling and Analysis*. McGraw-Hill, 2000
- [LT04] LI, Lijun ; TROPPER, Carl: Event Reconstruction in Time Warp. In: *Proceedings of the 18th Workshop on Parallel and Distributed Simulation*, 2004, S. 37–44
- [McC76] MCCABE, T. J.: A Complexity Measure. In: *IEEE Transactions of Software Engineering* SE-2.4 (1976), S. 308–320
- [MD03] MUECK, Bengt ; DITTMANN, Nico: *Marktanalyse: Materialfluss-Simulatoren*. ALB-HNI-Verlagsschriftenreihe, 2003 (11)
- [MDL⁺04] MUECK, Bengt ; DANGELMAIER, Wilhelm ; LAROQUE, Christoph ; FISCHER, Matthias ; KORTENJAN, Michael: Guidance of Users in Interactive 3D-Visualisations of Material Flow Simulations. In: SCHULZ, Thomas (Hrsg.) ; SCHLECHTWEIG, Stefan (Hrsg.) ; HINZ, Volkmar (Hrsg.): *Simulation and Visualisation 2004*, SCS European Publishing House, 2004, S. 73–84
- [Mer05] MERKURYEVA, Galina: Metamodelling for Simulation Applications in Production and Logistics / Department of Modelling and Simulation, Riga Technical University. 2005. – Forschungsbericht
- [MPK06] MARTENS, Jurgen ; PUT, Ferdi ; KERRE, Entienne: A Fuzzy Set Theoretic Approach to Validate Simulation Models. In: *ACM Transactions on Modeling and Computer Simulation* 16 (2006), October, Nr. 4, S. 375–398
- [MSJ93] MERTINS, K. ; SÜSSENGUTH, W. ; JOCHEM, R: *Modellierungsmethoden für rechnerintegrierte Produktionsprozesse*. Carl Hanser Verlag, 1993
- [Mue05] MUECK, Bengt: *Eine Methode zur benutzerstimulierten detaillierungsvarianten Berechnung von diskreten Simulationen von Materialflüssen*, University of Paderborn, Diss., 2005
- [Pal92] PALMORE, J.: Variable Resolution Modeling in Mathematics. In: DAVIS, P. K. (Hrsg.) ; HILLESTAD (Hrsg.): *Proceedings of the Conference on Variable-Resolution Modeling*, 1992
- [PBC98] PIDD, M. ; BAYER CASTRO, R.: Hierarchical Modular Modelling in Discrete Simulation. In: MEDEIROS, D.J. (Hrsg.) ; WATSON, E.F. (Hrsg.) ; CARSON, J.S. (Hrsg.) ; MANIVANNAN, M.S. (Hrsg.): *Proceedings of the 1998 Winter Simulation Conference*, 1998
- [PCG00] PANAYIOTOU, Christos ; CASSANDRAS, Christos ; GONG, Wei-Bo: Model Abstraction for Discrete Event Systems using Neural Networks and Sensitivity Information. In: *Proceedings of the 2000 Winter Simulation Conference*, 2000, S. 335–341

- [Pot97] POTHEN, Alex: Partitioning Algorithms with Applications to Scientific Computing / Old Dominion University. 1997. – Forschungsbericht
- [Pre01] PREIS, Robert: *Analyses and Design of Efficient Graph Partitioning Methods*, Universität Paderborn, Diss., 2001
- [PS04] PETTIE, Seth ; SANDERS, Peter: A Simpler Linear Time $2/3-\epsilon$ Approximation for Maximum Weight Matching. In: *Information Processing Letters* 91 (2004), S. 271–276
- [RCRS07] R. COBB, Barry ; RUMÍ, Rafael ; SALMERÓN, Antonio: Bayesian Network Models with Discrete and Continuous Variables. In: *Studies in Fuzziness and Soft Computing* 214 (2007), S. 81–102
- [RNT03] ROSER, Christoph ; NAKANO, Masaru ; TANAKA, Minoru: Buffer Allocation Model Based on a Single Simulation. In: *Proceedings of the 2003 Winter Simulation Conference*, 2003, S. 1238–1246
- [Ros99] ROSE, Oliver: Estimation of the cycle time distribution of a wafer fab by a simple simulation model. In: *Proceedings of the SMOMS 1999*, 1999, S. 133–138
- [Ros00] ROSE, Oliver: Why do simple Wafer Fab Models fail in certain Scenarios? In: *Proceedings of the 2000 Winter Simulation Conference*, 2000, S. 1481–1490
- [Ros07a] ROSE, Oliver: Improved Simple Simulation Models for Semiconductor Wafer Factories. In: *Proceedings of the 2007 Winter Simulation Conference* 1709–1712, 2007
- [Ros07b] ROSE, Oliver: Improving the Accuracy of Simple Simulation Models for Complex Production Systems. In: *Proceedings of the 2007 INFORMS Simulation Society Research Workshop*, 2007
- [Saa97] SAAB, Youssef: Combinatorial Optimization by Dynamic Contraction. In: *Journal of Heuristics* 3 3 (1997), S. 207–224
- [Sar05] SARGENT, Robert G.: Verification and Validation of Simulation Models. In: *Proceedings of the 2005 Winter Simulation Conference*, 2005, S. 130–143
- [Sch83] SCHRUBEN, Lee: Simulation Modeling with Event Graphs. In: *Communications of the ACM* 26 (1983), Nr. 11, S. 957–963
- [Sev90] SEVINC, Suleyman: Automation of Simplification in Discrete Event Modelling and Simulation. In: *International Journal of General Systems* 18 (1990), Nr. 2, S. 125–142
- [Sev91] SEVINC, Suleyman: Theories of Discrete Event Model Abstraction. In: NELSON, B. L. (Hrsg.) ; KELTON, W. D. (Hrsg.) ; CLARK, G. M. (Hrsg.): *Proceedings of the 1991 Winter Simulation Conference*, 1991, S. 1115–1119

- [SFH⁺08] SUESS, Tim ; FISCHER, Matthias ; HUBER, Daniel ; LAROQUE, Christoph ; DANGELMAIER, Wilhelm: A System for Aggregated Visualization of Multiple Parallel Discrete Event Simulations. In: *International Symposium on Advances in Parallel and Distributed Computing Techniques (APDCT-08)* IEEE, IEEE Computer Society Press, Dezember 2008, S. 587–593
- [SMM07] SUNARJO, Meng F. ; MEINHARDT, Ingolf ; MARQUARDT, Hans-Georg: Dimensionierung von Entkopplungspuffern in dynamischen Fertigungsprozessen mittels Warteschlangen / Fraunhofer IML. 2007. – Nichtreferierte Veröffentlichung im Logistics Journal ISSN 1860-5923
- [SR96] SKÖLD, Sven ; RÖNNGREN, Robert: Event Sensitive State Saving in Time Warp Parallel Discrete Event Simulation. In: *Proceedings of the 1996 Winter Simulation Conference*, 1996, S. 653–660
- [SW00] SCHUMACHER, R. ; WENZEL, S.: Der Modellbildungsprozess in der Simulation. In: *Referenzmodelle für die Simulation in Produktion und Logistik*, SCS-Europe, 2000, S. 5–11
- [SY88] SCHRUBEN, Lee W. ; YÜCESAN, Enver: Simulation Graphs. In: ABRAMS, M. (Hrsg.) ; HAIGH, P. (Hrsg.) ; COMFORT, J. (Hrsg.): *Proceedings of the 1988 Winter Simulation Conference*, 1988, S. 504–508
- [SY93] SCHRUBEN, Lee ; YÜCESAN, Enver: Complexity Of Simulation Models: A Graph Theoretic Approach. In: EVANS, G. W. (Hrsg.) ; MOLLAGHASEMI, M. (Hrsg.) ; RUSSEL, E. C. (Hrsg.) ; BILES, W. E. (Hrsg.): *Proceedings of the 1993 Winter Simulation Conference*, 1993
- [VAD⁺08] VÖCKLING, Berthold ; ALT, Helmut ; DIETZFELBINGER, Martin ; REISCHUK, Rüdiger ; SCHEIDELER, Christian ; VOLLMER, Heribert ; WAGNER, Dorothea: *Taschenbuch der Algorithmen*. Springer, 2008
- [VB05] VAN BEERS, Wim C.: Kriging Metamodeling in Discrete-Event Simulation: An Overview. In: *Proceedings of the 2005 Winter Simulation Conference*, 2005, S. 202–208
- [VDI73] VDI. *Materialfluß-Untersuchungen*. VDI-Richtlinien. 1973
- [VDI96] VDI. *VDI-Richtlinie 3633 Blatt 1: Simulation von Logistik-, Materialfluss- und Produktionssystemen Begriffsdefinition*. VDI-Richtlinien. November 1996
- [VDI00] VDI. *VDI-Richtlinie 3633 Blatt 1: Simulation von Logistik-, Materialfluss- und Produktionssystemen Grundlagen*. VDI-Richtlinien. März 2000
- [VG03] VÖLKER, Sven ; GMILKOWSKY, Peter: Reduced Discrete-Event Simulation Models for Medium-Term Production Scheduling. In: *Systems Analysis Modelling Simulation* 43 (2003), S. 867–883
- [Völ03] VÖLKER, Sven: *Reduktion von Simulationsmodellen zur simulationsbasierten Optimierung in der Termin- und Kapazitätsplanung*. Peter Lang Europäischer Verlag der Wissenschaften, 2003

- [Wal87] WALLACE, Jack C.: The control and transformation metric: toward the measurement of simulation model complexity. In: *WSC '87: Proceedings of the 19th conference on Winter simulation conference*, ACM Press, 1987, S. 597–603
- [Zei76] ZEIGLER, Bernhard P.: *Theory of Modeling and Simulation*. Academic Press, 1976
- [ZPK00] ZEIGLER, Bernhard P. ; PRAEHOFER, Herbert ; KIM, Tag G.: *Theory of Modeling and Simulation*. 2nd Edition. Academic Press, 2000

Stichwortverzeichnis

- d³FACT , 56
- Anforderungen, 18
- Ausgangsmodell, 7, 73
- Belegung, 121
- Channel, 28, 62
- DEVS, 29
- Durchlaufzeit, 51, 115
- Durchsatz, 115
- dvdS, 9, 26, 55, 57, 64
- Engstelle, 87
- Ereignis, 7, 119
 - routine, 7, 62
 - Routine, 29
- Erzeugungsbeziehung, 120
- Hierarchie, 10
- Komplexität, 8, 49
 - Komponentenklasse, 110
 - Maß, 49, 50, 113
 - Ziel-, 109
- Komponente, 84
 - Förderband, 82
 - Klasse, 77, 84
 - Maschine, 80
 - Puffer, 82, 104
 - Quelle, 78
 - Senke, 80
 - Sequenz, 82
 - Verteiler, 83
 - Zusammenführer, 83
- Laufzeit, 8
- Materialfluss, 6, 58
- Modell, 6
 - beschreibung, 26, 28, 76, 84
 - Verhalten, 51, 114
- Partitionierung, 30, 73, 84
 - Greedy, 32
 - Kernighan-Lin, 31
 - Multilevel, 33
 - Zielfunktion, 85
- Ratio-Cut, 85
- Regelung, 14
 - Algorithmus, 116
- Rekomposition, 10, 57, 109
- relative effektive Kapazität, 87
- Replikation, 114
- Simulation, 6
 - diskrete, 6
 - Konsistenz, 54
 - Materialfluss, 7, 77
 - Zustand, 15, 29, 52, 84, 119
- Simulationskernel, 60
- Störung, 88, 123
- Struktur, 58, 89
- System, 6
- Teilmodell, 10, 73, 84
 - Zustand, 119
- Testmodell, 129
- Umlaufbestand, 51, 115
- Vereinfachung, 8, 86
 - geregelte, 15, 76, 116
 - Regel, 91, 98, 104
 - Variablentransformation, 102

- Weglassung, 103
- Zusammenfassung, 87, 90
- Verfügbarkeit, 88
- Verhaltensabweichung, 9, 113
 - Maß, 115
 - Ziel-, 109
- Verknüpfung, 84
 - Gewichtung, 86
- Versuchsplan, 127
- WIP, 115
- Zielvorgaben, 109
- Zufallsverteilung, 78
- Zustandsabbildung, 16, 73, 120
- Zustandsvariable, 84, 119
- Zyklus, 50, 87
 - quotient, 111